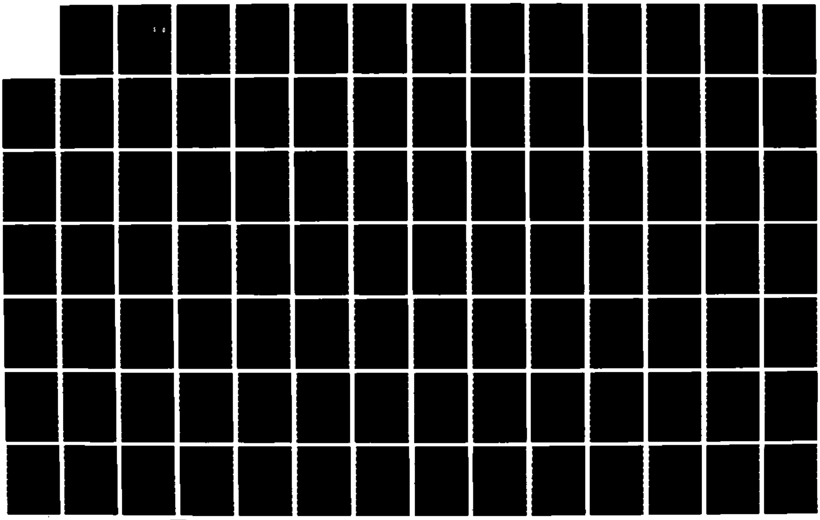
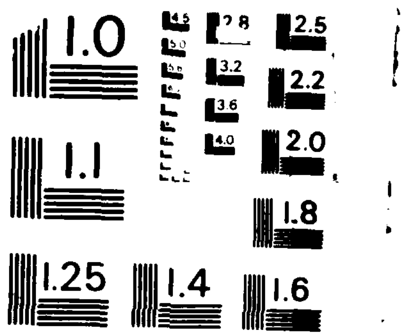
AD-A176 909 PROJECT BIDDING UNDER CHANCE TIME ESTIMATES(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH R S VOGTMANN 1986 AFIT/CI/NR-87-32T 1/2

UNCLASSIFIED F/G 5/1 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A176 909

Project Bidding Under Chance
Time Estimates

by
Russell S. Vogtmann

A thesis submitted to the Graduate Faculty
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Department of Operations Research

Raleigh
1986

Approved by:

Chairman of Advisory Committee

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/CI/NR 87-32T	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Project Bidding Under Cahnce Time Estimates		5. TYPE OF REPORT & PERIOD COVERED THESIS/~~DISSERTATION~~
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Russell S. Vogtmann		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS AFIT STUDENT AT: North Carolina State Univ		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS AFIT/NR WPAFB OH 45433-6583		12. REPORT DATE 1986
		13. NUMBER OF PAGES 179
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

APPROVED FOR PUBLIC RELEASE: IAW AFR 190-1

Lynn Wolaver 30 JAN 87

LYNN E. WOLAVER
Dean for Research and
Professional Development
AFIT/NR

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

ATTACHED

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Abstract

VOGTMANN, RUSSELL S. 2nd Lt., U.S. Air Force, 1987. Project Bidding Under Chance Time Estimates (Under the direction of Salah E. Elmaghraby.) 76pp. Master of Science, North Carolina State University.

The bidding for the contract of a project is a very important phase in the life of the project. In today's competitive market for a project contract, it is essential to have accurate information in order to make a bid that will yield a profit and still win the contract.

~~In~~ this thesis ~~we~~ examine projects that can be modeled by directed acyclic networks under probabilistic activity durations and costs (the PERT model). For a successful bid, the contractor must recognize that there are two streams of cash flow: an "outflow" caused by the undertaking of the activities of the project, and an "inflow" due to income received at the time of realization of pre-designated "key events". Under the PERT model of activity networks, both streams of cash flow are subject to uncertainty. The contractor's problem then becomes that of the estimation of these two streams at sufficient confidence to realize a reasonable profit for him.

A Monte Carlo simulation is developed for solution of the problem, and a computer package has been constructed to assist project managers in the determination of the bid package.

Table of Contents

1. INTRODUCTION

The process of bidding for the contract of a particular project is a very important phase in the life of that project. And although bidding (and its precursor, cost estimation) has been a topic of research for decades, it has been largely overlooked in the area of activity networks (ANs). This fact seems odd since from the very beginnings of PERT (Program Evaluation and Review Technique), the Department of Defense has demanded the use of the PERT model of the project in any proposal submitted to it [6]. If the contractor, in his proposal, must submit a network detailing the method by which he plans to complete the project, then it would make sense that he use that information in the preparation of his bid on the project. Presently, for both the contractor and the owner of a project, project bids are typically determined on the basis of 'averages' based on past performance. As a consequence, a project that represents a novel undertaking in either its technological content or in its scope poses a dilemma to both sides, with the blind leading the blind in the ensuing negotiations. And yet, it is precisely these unique features that gave rise to the concept of 'project management' in the first place! [7].

Clearly this method fails to use the information provided in the network model of the project. The reason for this discrepancy, however, does not lie in the hands of the

project manager. The research in the theory of ANs (specifically, directed acyclic networks), since their inception in 1959, has been mainly directed in the following areas [5]: 1) optimal project "compression", 2) optimal planning and acquisition of resources, 3) optimal scheduling under constrained availability of scarce resources, and 4) analysis and synthesis of projects under conditions of uncertainty. Of these four areas, only the first has dealt with cost considerations, and then only in the deterministic case. So we see the theory has not yet evolved sufficiently to be successfully applied to the area of project bidding.

The literature of activity networks has long recognized the need to somehow account for the cost incurred due to the activities. In 1971, Kleindorfer [10] discussed three money flow problems due to an activity: 1) Flows that accrue at the start or end of an activity, 2) Time-varying flows that accrue during the execution of an activity, and 3) Time-varying flows that begin at the start or end of an activity and continue thereafter. These cost considerations, however, have mainly manifested themselves in cost control [5] rather than in determining a project bid prior to the inception of the project. There have been few instances in which an attempt has been made to utilize AN theory to determine that bid.

In 1972, Case [2], in a paper titled "On the Consideration of Variability in Cost Estimating", made an attempt to

determine a cost distribution by using a three estimate approach to estimate activity cost similar to a PERT estimate of activity duration. Then the project cost is the sum of these activity cost random variables (which by appealing to the Central Limit Theorem is appoximately normally distributed), from which a mean and variance of project cost could be determined and various probabilistic statements could be made. Unfortunately, this simplistic method falls far short of providing any realistic help in dealing with the problem at hand. In 1974, in a paper titled "Interrelating Project Estimates to CPM Schedules", Badger [1] discussed a method of cross referencing estimate accounts to network activities to provide compatibility between the two project control documents. The approach goes into detail on the allocation of the project resources to the specific activities that require them. This appears to be the first place in the literature that directly uses the project network when making project estimates. The remaining literature effectively ignores the relationship between the project network and project bidding.

It is obvious that the process of infusing the theory of activity networks into the formulation of a project bid has not been adequately addressed to this date, especially in the case of PERT networks.

2. PROBLEM STATEMENT

When faced with preparing a bid to submit to the owner of a project, the contractor must analyze the project to determine how he will complete the project. Included in this analysis is the formulation of a project network and the determination of the resources (and consequently costs) that are required to complete it. The contractor will often define various key events in the project at which he will require the owner to make partial payments for work that has been completed, and often requires an initial payment before the work on the project begins.

Farid and Boyer [8] indicate that the bid that results should be the total cost of the project multiplied by a "fair and reasonable markup" (FaRM) of the project cost. The FaRM is viewed as the smallest markup which satisfies the Required-Rate-of-Return (RRR) of the contractor for the particular project at hand. A FaRM of a project, given a RRR, can be obtained via AN theory by analyzing the two streams of cash flow inherent in any project. These streams are: 1) "outflow": payments made by the contractor in executing the project, and 2) "inflow": receipts from the owner of the project at various key events (i.e., those events at which payments are made to the contractor). In the case where all activities in the network can be modeled deterministically (in duration and cost), there is no problem in defining the cost of an activity, and subsequently

the cost of the project. The two streams of cash flow can then be easily obtained given a particular schedule of the network activities, and a FaRM determined. An illustration of this process is presented in section 3.

Russell [13] has formulated a mathematical program to determine the schedule of activities that maximizes the net present value of a deterministic network given cash flows and resource restrictions. Several heuristics were compared for solution of the nonlinear program. The contractor will then have an optimal schedule by which he performs the activities of the project.

Unfortunately, when we are dealing with PERT-type stochastic networks (characterized by activity durations and costs that are both probabilistic), the realization time of a key event as well as the costs of the activities leading to the realization of the key event can only be determined in a probabilistic sense. At this point, the manager of the project must make a decision on the amount of risk he wants to take in the process of setting a delivery date and the cost of the subgraph of the key event. Further complicating the issue of obtaining a FaRM in PERT networks is the problem of scheduling the activities necessary to calculate the stream of cash "outflow".

When these problems are adequately addressed in the case of a PERT network, we will have a logical and theoretically sound method of determining a bid package for projects

that can be modeled as directed acyclic networks. This method would appear to be superior to methods that are presently used to determine bids in industry. Whether it will be practical in real-life situations has yet to be determined.

Purpose of Research

The purpose of this research is to address the problem of determining a bid package -- i.e., the payment amounts and dates, for projects that can be modeled by a directed acyclic network.

3. A DETERMINISTIC EXAMPLE

The simplest form of the bidding problem in the context of ANs lies in the deterministic (CPM) model. This model is a first step to solving the more general probabilistic model. In this case, it is easy to determine the two streams of cash flow, and consequently calculate the FaRM given a RRR. The following example will introduce some necessary concepts of the bidding problem, which will then be generalized to the probabilistic model that is the main concern of this thesis.

Assume it is desired to bid on a project that is modeled with the data in Table 1 and the network in Figure 1.

Table 1. Activity Durations and Costs for CPM Example.

	Activity	Duration	Cost/day	Fixed Cost
1.	Excavate	2	100	500
2.	Foundation	4	300	1000
3.	Rough wall	10	250	1000
4.	Rough ext. plumbing	4	150	200
5.	Roof	6	250	1000
6.	Rough interior plumbing	5	150	200
7.	Rough electrical work	7	100	200
8.	Exterior siding	7	75	200
9.	--dummy--	0	0	0
10.	Exterior painting	9	40	50
11.	Wall board	8	40	25
12.	Interior painting	5	40	50
13.	Flooring	4	75	100
14.	--dummy--	0	0	0
15.	Exterior fixtures	2	200	100
16.	Interior fixtures	6	100	300

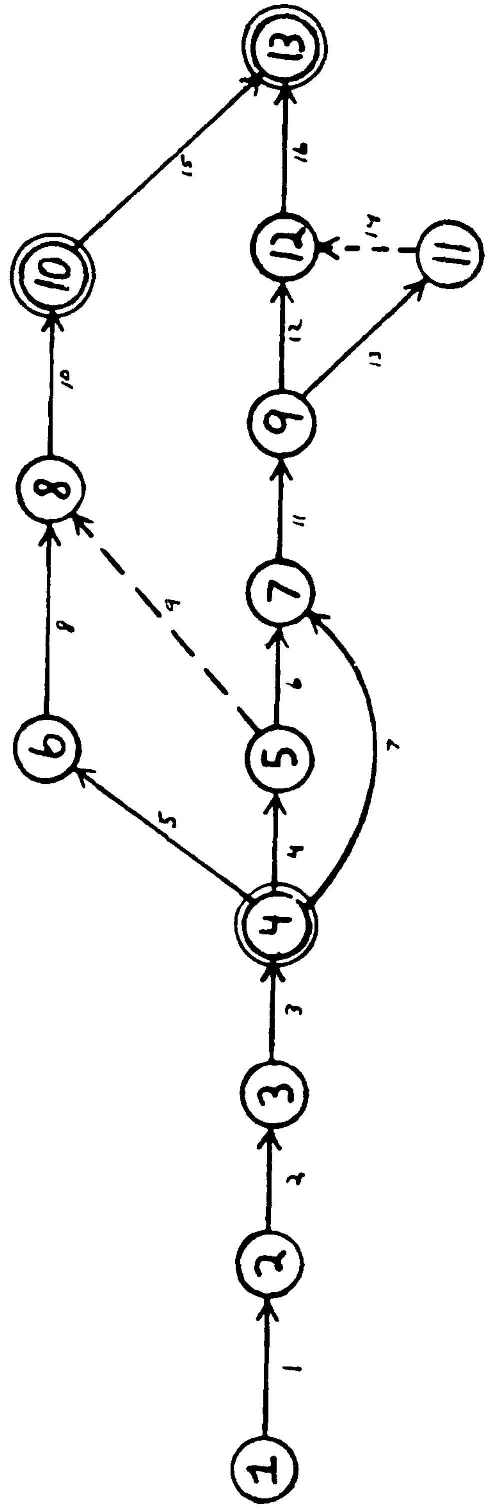


Figure 1. Network for Deterministic Example.

Table 2. Bar Chart and Cash "Outflow" for CPM Example.

```
--------------------------------------------------------
Act Dur   .    .    .    .    .    .    .    .    .    .
--------------------------------------------------------
1   2     |--
2   4     |  ----
3   10    |      ----------
4   4     |                ----
5   6     |                ------
6   5     |                    -----
7   7     |                -------
8   7     |                      -------
9   0     |
10  9     |                             ---------
11  8     |                         --------
12  5     |                                 -----
13  4     |                                 ----
14  0     |
15  2     |                                      --
16  6     |                                      ------
--------------------------------------------------------
"outflow" |2600|2300|1250|3650|2175| 615| 700|1390| 400|
```

The key events of the project have been pre-determined to be 4, 10, and 13, and for this example assume that all activities are scheduled to begin at their earliest start times. (The implications of activity float in the deterministic model of this problem will be discussed near the end of the example). The bar chart in Table 2 shows when each activity is scheduled to be performed, as well as the cost incurred per period (assume a 'period' length of 5 'units'; e.g., a period represents a week of 5 days). Note that in this deterministic example, the activity schedule is very well defined. The cost incurred in a period is calculated by adding the fixed costs of activities that begin in the period plus the cost/day of the activities performed in the period times the number of days that they are performed in that period. To illustrate, the calculations for the cash "outflow" in periods 1 and 7 are shown.

<u>period 1</u>

activity	fixed cost	cost/day	# days	total cost
1	500	100	2	700
2	1000	300	3	1900

cost incurred in period 1 = 2600

<u>period 7</u>

activity	fixed cost	cost/day	# days	total cost
10	---	40	5	200
11	---	40	3	120
12	50	40	2	130
13	100	75	2	250

cost incurred in period 7 = 700

To calculate the cash "inflow" of the project, we must determine two elements: (i) the cost associated with the subgraph of each key event, which is then multiplied by some 'markup' to obtain the cash "inflow", and (ii) the time of realization of the key event. The cost of the subgraph to the key event is the sum of the costs of the activities in the subgraph. These calculations are displayed below and summarized in Table 3.

key event 4

activity	fixed cost	cost/day	# days	total cost
1	500	100	2	700
2	1000	300	4	2200
3	1000	250	10	3500

cost of subgraph = 6400

key event 10

activity	fixed cost	cost/day	# days	total cost
4	200	150	4	800
5	1000	250	6	2500
8	200	75	7	725
9	0	0	0	0
10	50	40	9	410

cost of subgraph = 4435

key event 13

activity	fixed cost	cost/day	# days	total cost
6	200	150	5	950
7	200	100	7	900
11	25	40	8	345
12	50	40	5	250
13	100	75	4	400
14	0	0	0	0
15	100	200	2	500
16	300	100	6	900

cost of subgraph = 4245

Table 3. Cost of Subgraphs and Realization Times for CPM Example.

Key event	cost of subgraph	realization time
4	6400	16
10	4435	38
13	4245	44

Table 4 displays the cash flows for the project given a zero markup of cost for the cash "inflow".

Table 4: Cash Flows for CPM Example (assuming a zero cost markup, and the percentage of the project cost that we require at the beginning of the project is 0%).

Period	1	2	3	4	5	6	7	8	9
"outflow"	2600	2300	1250	3650	2175	615	700	1390	400
"inflow"				6400				4435	4245

Now, given this information, we should be able to calculate the FaRM for this project for a given RRR. For the sake of this example, let RRR=20%, initial capital=$500, interest rate on money deposited=10%, interest rate on money borrowed=15%, and the retention percentage=12%. Then if we perform the required calculations of simple financial management (i.e. paying the cash "outflow", receiving the cash "inflow", depositing money at an interest rate, and borrowing money at an interest rate when applicable), we obtain a FaRM=1.2051. The pertinent results of the calculations are displayed in table 5 where it is assumed that the cash "out-

flow" is incurred at the beginning of the period and the cash "inflow" occurs at the end of the period (e.g., the cash "outflow" in period 5 and the cash "inflow" in period 4 occur simultaneously in the calculations at the beginning of period 5) This would indicate that we make a bid of \$18,247.22 in this project which has a cost of \$15,141.17.

The progress payments can be easily determined using the FaRM and the cost of the subgraphs. These progress payments are \$7721.80, \$5371.18, \$5153.65, respectively.

Table 5. FaRM Calculations for Deterministic Example

FaRM (Activity Worth / Activity Cost) = 1.2051

Period	Interest*	Cash Balance**	Retention#	Cum Ret.##
0	-----	500.00	----	----
1	(4.02)	-2104.02	0.00	0.00
2	(8.45)	-4415.62	0.00	0.00
3	(10.86)	-5679.91	0.00	0.00
4	(17.91)	-9362.82	0.00	0.00
5	(9.11)	-4763.44	926.65	927.86
6	(10.32)	-5392.97	0.00	929.07
7	(11.69)	-7528.14	0.00	931.50
9	(6.14)	-3211.87	644.56	1578.12
10	1.73	1325.21	618.46	2199.45
11	1.73	1326.94	0.00	2202.32

Terminal Cash Position### = 3529.26
Profit = .200

*Interest on money borrowed is in parenthesis.
**Cash balance at end of period.
#Amount retained at beginning of period.
##Cumulative retention at end of period.
###Terminal cash = cash balance + cum.retention.

Activity Float Considerations

The above FaRM was obtained assuming that all activities begin at their earliest start times. Of course, in the deterministic model, all activities that are not contained in the critical path have float. Therein lies the only possible flexibility in the bidding strategy for the deterministic model. Logic dictates that the contractor would want to start all activities as late as possible (thereby delaying his disbursements) without causing a delay in the realization times of the key events (and subsequently delaying his cash "inflow"); see Russell [13].

Note that if the contractor has overhead that is incurred from the beginning to the end of the project, he can simply add another arc in the network from the start node to the terminal node indicating the appropriate cost/unit time. This process of adding arcs to the network can be used to model costs that are inherent in the execution of any project (be it overhead cost or otherwise). The network of Figure 2 illustrates this point. If we add an arc from node 1 to node 13 with a duration of 44, a fixed cost of zero and a cost/day of 25.00 for administration plus 35.00 for overhead cost, and a zero duration arc from node 13 to node 14 to model some fixed cost of 500.00 incurred at the conclusion of the project, the network in Figure 2 results. Assuming the same values for capital, RRR, etc., one obtains a FaRM= 1.2055. This would result in a bid of $22,061.70 for the

project that has a cost of $18,300.79.

This deterministic example displays the cash flow concepts involved in the bidding problem. When we generalize this model to include activities and costs that are probabilistic in nature, we introduce complications in the realization time and cost of subgraphs of key events, as well as in determining the cash "outflow" of the project.

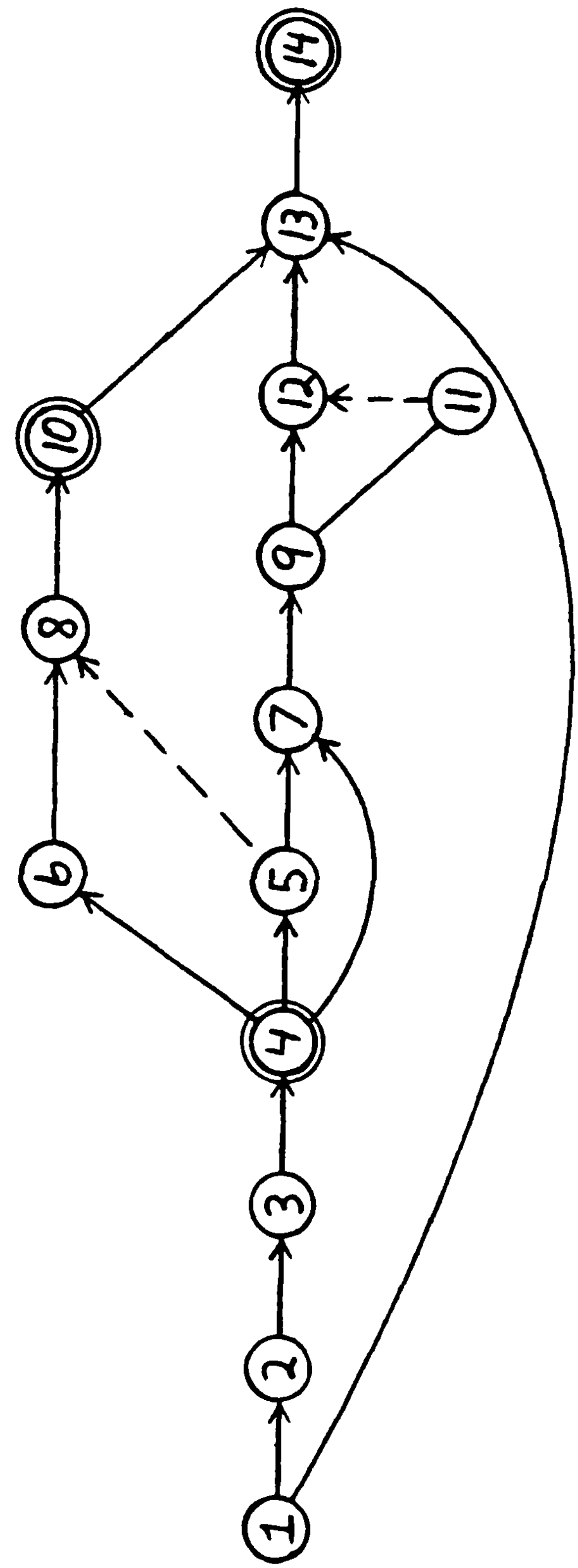


Figure 2. Network of Fig.1 with Added Costs

4. METHODOLOGY

From this point on, when referring to the bidding problem, we assume the structure of the AN is known. From a theoretical point of view, the major problem is the determination of the appropriate method of separation of cost of activities that belong to more than one key event. To this end, we examine the subgraphs of the various key events of a project. Define a subgraph of a node in a network as the subnetwork terminating at that node. If we define S(i) = {set of activities belonging to the subgraph of key event i}, then for key events i and j, either (a) S(i) and S(j) are disjoint (no common activities), (b) S(i) is a subset of S(j) or S(j) is a subset of S(i), or (c) S(i) and S(j) overlap (i.e. they have elements in common, but neither is a subset of the other). Figure 3 graphically displays these situations.

4.1 Cost Separation of Common Activities

When S(i) and S(j), for all $i \neq j$, are disjoint, the cost of each activity is assigned to its respective key event.

When S(i) is a subset of S(j), the cost of the subgraph of S(i) is assigned to key event i, and the cost of the remaining activities is assigned to key event j.

If S(i) and S(j) overlap, a situation exists in which an activity is in the subgraph of more than one key event

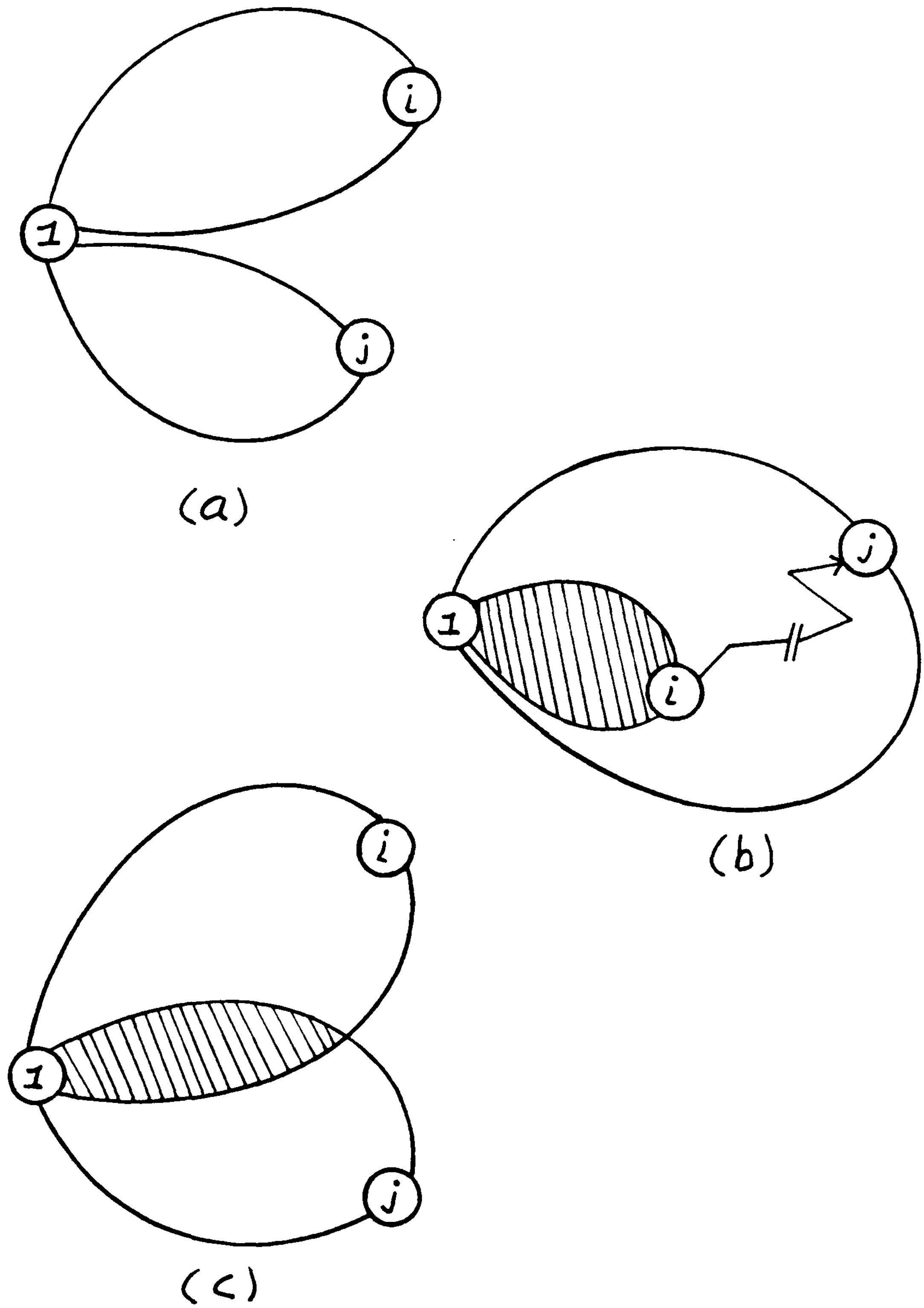


Figure 3: Relationships between Subgraphs of Key Events

and neither is a subset of the other. The simple network example in Figure 4 (with key events 4, 5, and 6) will illustrate these concepts: activity 'a' is common to all key events, activity 'c' is common to key events 4 and 6, activities 'b', 'd', and 'e' are common to key events 5 and 6, and activities 'f' and 'g' are unique to key event 6. Since activities 'f' and 'g' are unique to key event 6, their cost is attributed to that key event. The costs of the remaining activities are assigned to either key event 4 or 5. Clearly, the cost of activity 'c' is assigned to key event 4, and the costs of activities 'b', 'd', and 'e' are assigned to key event 5.

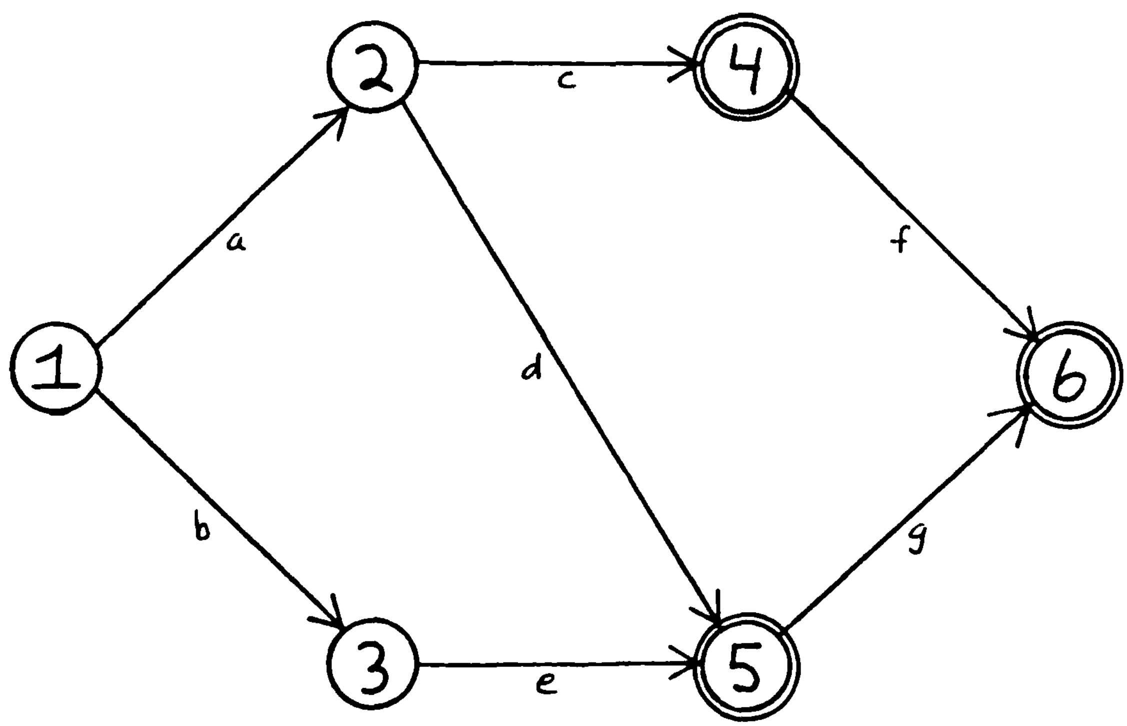


Figure 4: Network Illustrating Common Activities

Since activity 'a' is common to 4 and 5, we must either assign the cost of activity 'a' to one of the key events or divide its cost, attributing a percentage of its cost to each key event in such a manner so there is no "double costing" for the same activity. The contractor would want to receive the payment for common activities as early as possible, thereby maximizing his profit, and the owner would rather pay as late as possible, thus minimizing his expense. We wish to address this issue objectively and obtain a logical and consistent method of separating the costs of the common activities.

There are many heuristics that could be used to address this problem. The following are just a few:

Heuristic 1. Separate costs based on the probability of the earliest realization of the key events.

Heuristic 2. Separate costs based on the activity criticality index in each key event subnetwork.

Heuristic 3. Separate costs based on the relative cost of the subgraphs of the key events.

Heuristic 4. Separate costs based on the criticality index of the key events themselves.

In section 4.3 we criticize heuristics 2-4, but we first develop an argument supporting heuristic 1. Since the key event that is realized first requires the common activities to be completed by its realization time, it is reasonable to assign the cost of the common activities (or a large

portion of it) to that key event. In mathematical terms, it seems reasonable to attribute a greater share of the cost of common activities to those key events having the greatest probabilities of being realized first. Note that in a deterministic network, this would translate into attributing the entire cost of common activities to the key event that is realized at the earliest time.

To illustrate, we continue the analysis of the network in Figure 4. The subgraphs of key events 4 and 5 are shown in Figure 5. Assume that all activity durations are distributed as a Gamma random variable with parameters (t=2, lambda=1). Then, the realization time distributions, means, and variances are as listed in Table 6. The calculations required to determine the distribution functions are summarized in table 7.

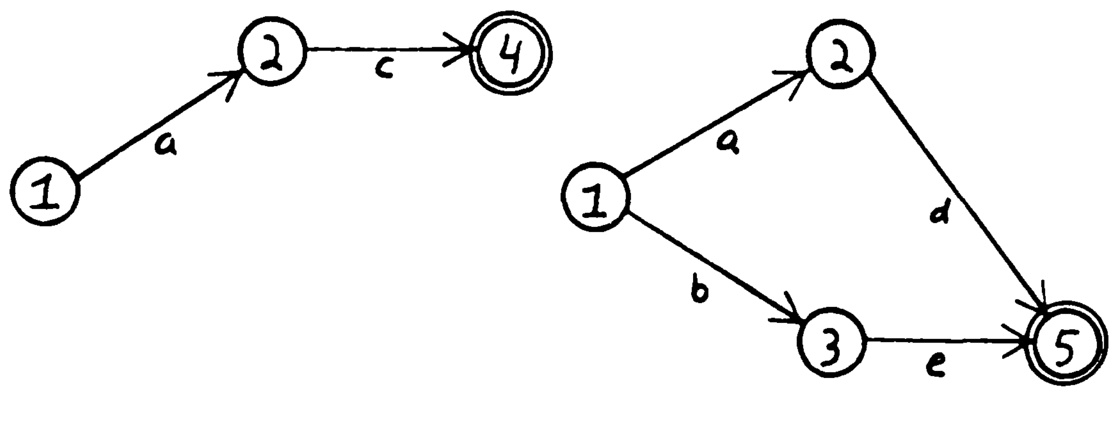


Figure 5: Subgraphs of Key Events 4 and 5

Table 6: Parameters of key events 4 and 5 of Figure 4.

Node 4: $T_4 = Y_a + Y_b$

$F_4(t) = 1 - \exp(-t)*(1 + t + t^2/2 + t^3/6)$

Mean = 4
Variance = 4

Node 5: $T_5 = \max\{ Y_a + Y_d \; ; \; Y_b + Y_e \}$

$$F_5(t) = 1 - 2*\exp(-t)*(1 + t + t^2/2 + t^3/6)$$

$$+ \exp(-2*t)*(1 + t + t^2/2 + t^3/6)^2$$

Mean = 5.46
Variance = 4.17

Table 7: Realization Time Distribution Calculations for Example Network

Random Variable	Representation	Probability Distribution Function
T_1	0	
T_2	Y_a	$1-\exp(-t)*(1-t)$
T_3	Y_b	$1-\exp(-t)*(1-t)$
T_4	T_2+Y_c	$1-\exp(-t)*(1+t+t^2/2+t^3/6)$
W_1	Y_a+Y_d	$1-\exp(-t)*(1+t+t^2/2+t^3/6)$
W_2	Y_b+Y_e	$1-\exp(-t)*(1+t+t^2/2+t^3/6)$
T_5	$\max(W_1,W_2)$	$1-2*\exp(-t)*(1+t+t^2/2+t^3/6)$
		$+\exp(-2*t)*(1+t+t^2/2+t^3/6)^2$

The probability density functions of the node realization times are graphed in Figure 6.

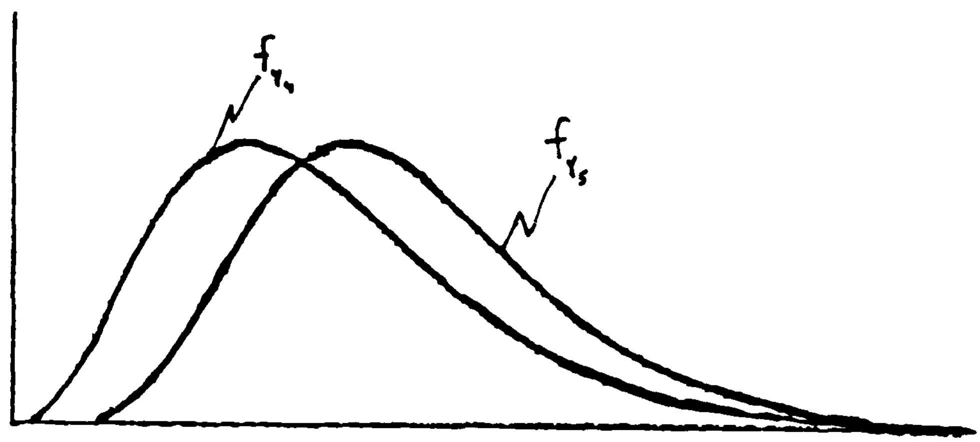


Figure 6: Density Functions of Key Events 4 and 5

If we let Y_4 denote the realization time of key event 4 and Y_5 denote the realization time key event 5, then from the data it is clear that $Pr(Y_4 \leq Y_5)$ will be relatively high (the allocation calculations will be presented in section 4.2). Therefore, we will allocate a greater percentage of the cost of activity 'a' to key event 4.

In general, the allocation percentage of key event i would be calculated as follows: let Y_i denote the realization time of key event i (the key event for which we desire to determine the cost allocation percentage), and Y_j denote the realization time of key event j, for $j \epsilon H(i)$, where H(i) is the set of key events that share activities with key event i, and let N denote the number of key events in H(i).

Furthermore, let A_j denote the event [$Y_i \leq Y_j$] (i.e. key event i is realized before key event j), for $j \epsilon H(i)$, and let Prop(i) denote the proportion of the cost of common activities allocated to key event i. Then,

$$\text{Prop}(i) = \Pr \{ A_1 , \ldots, A_N \}. \tag{1}$$

This yields the probability that key event i is realized first. Unfortunately, this probability statement is not immediately amenable to solution since the A_j's are not independent. To resolve this, we assume independence of the elements in H(i) and condition on the random variable Y_i. Let B_j denote the event [$Y_j \geq y$] (i.e. node j is realized after time y), for $j \epsilon H(i)$. Then since we assume that the Y_j's are independent,

$$\text{Prop}(i) = {}_0\int^{\infty} \Pr \{ B_1, \ldots, B_N \} * dF_Y(y) \tag{2}$$

$$= {}_0\int^{\infty} \Pr\{ B_1 \} * \ldots * \Pr\{ B_N \} * dF_Y(y) \tag{3}$$

Therefore, when we know the forms of the distribution functions of the realization times of the key events, we can use the above equation to solve for the cost allocation percentages.

4.2 Method of Separating Costs of Common Activities

The algorithm presented next is based on the above argument. To facilitate the computations, we make the following assumptions: 1) realization times of key events are normally distributed (or can be thus approximated), and 2) the events A_j can be considered independent.

The computer software developed uses either this method, or permits the user to assign proportions as he sees fit.

Cost Separation Algorithm:

1) Denote the key event with the smallest mean by key event s, then determine the probability, $p(i)$, that key event s is realized before key event i, for $i \epsilon H(s)$.

$$p(i) = Pr\{ Y_s \le Y_i \} \qquad (4)$$

---if $p(i)$ is greater than or equal to 0.9, assign a proportion of zero to key event i. This is an arbitrary cut-off point.

2) all remaining key events are given percentages as follows:

---let $w(i) =$

$$= Pr\{ Y_i \le Y_1 \} * \ldots * Pr\{ Y_i \le Y_N \} \qquad (5)$$

---then, let the proportion $Prop(i)=w(i)/\Sigma_i w(i)$

Applying the above algorithm to the example network in Figure 4 yields:

$$p(5) = Pr\{ Y_4 \le Y_5 \} = .695$$

Therefore,

$$w(4) = .695, \quad \text{or} \quad Prop(4) = .695,$$

$$\text{and } w(5) = .305, \quad \text{or} \quad Prop(5) = .305.$$

Therefore, we allocate 69.5% of the cost of activity 'a' to key event 4, and 30.5% of the cost to key event 5.

To show the limiting case when the probability of realizing one key event earlier than another is very high, let

the network in Figure 4 be modified as follows: activity 'c' is distributed as a Gamma (t=20, lambda=1). The resulting distribution parameters are:

node 4: mean = 22, variance = 22,

and node 5: mean = 5.46, variance = 4.17.

The cost separation algorithm results in the following:

$$p(4) = \Pr\{ Y_5 \le Y_4 \} = .999$$

Therefore,

$$\text{Prop}(4) = 0.0, \text{ and } \text{Prop}(5) = 100.0$$

The algorithm assigns the entire cost of activity 'a' to key event 5, as intended.

4.3 An Alternative Cost Separation Method

Heuristic 2 suggests an alternate method of cost separation that considers the importance of the common activities to their respective key events. We examine it here to display a problem that it shares with heuristics 3 and 4. To illustrate, consider the network of Figure 7. Then the activities that are common to key events 4 and 5 are 'a', 'b', and 'd'. The criticality index of an activity is defined as the probability that the activity is on the critical path; see Elmaghraby [5], p. 277. Now, if we define the importance of the activity as the criticality of the activity in the subgraph of the key event, we can determine a percentage to assign to the key events.

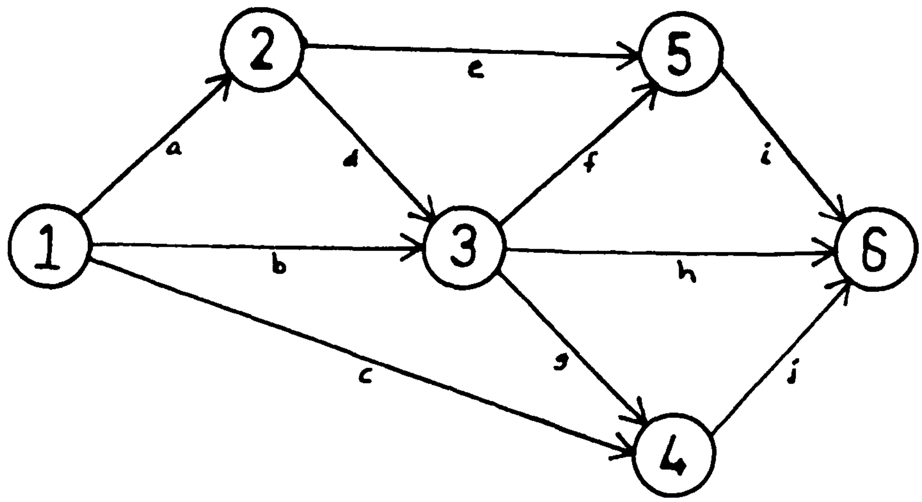


Figure 7: Network Example for Alternative Cost Separation Method

Figure 8 shows the key event subgraphs and the approximate activity criticality indices for the given activity duration distributions.

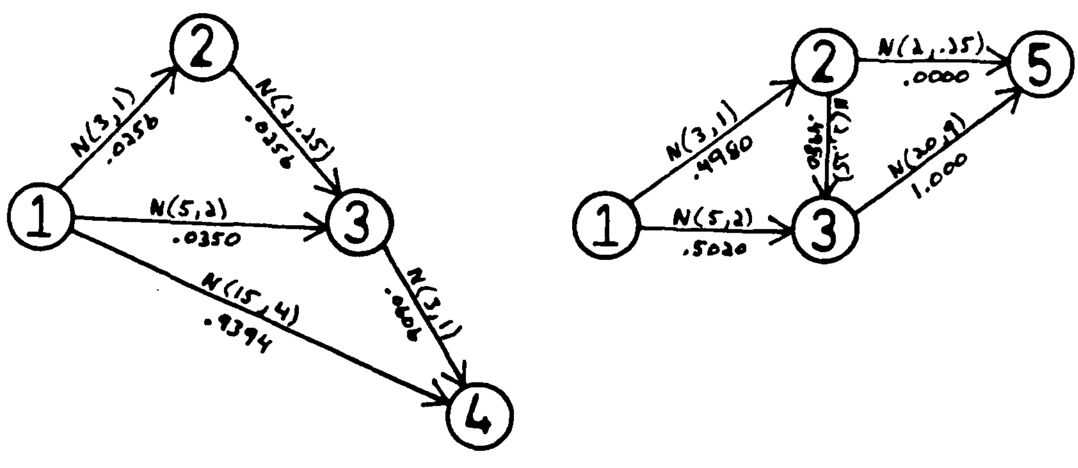


Figure 8: Subgraphs of Key Events 4 and 5 of Fig. 7

Activity 'b' has a criticality index of .035 in the subgraph of node 4, and .502 in the subgraph of node 5. This gives us an indication of the relative importance of activity 'b' to its key events from which we can determine a percentage of the cost allocation. Thus, according to the logic of this heuristic, the overwhelming majority of the cost of activity 'b' should be allocated to key event 5.

Although this method of separating costs is intuitively appealing, it ignores the fact that the total activity cost is incurred long before the key event, to which we have attributed a majority of the common activity cost, is realized. The network in Figure 7 illustrates this problem. We expect to realize node 4 before time 20.1 with probability .95, and expect to realize node 5 after time 23.1 with probability .95. Therefore, the total cost of activity 'b' is expected to be expended long before node 5 is realized, yet we do not account for most of it until node 5 is realized because of the method of cost allocation. Clearly, this is not in the best interest of the contractor who is trying to maintain liquidity. Also, in a PERT network, determining the criticality index of an activity, or even approximating it, is a difficult problem; see Dodin and Elmaghraby [4].

4.4 Distributions Associated With Key Events

Before a bid proposal can be submitted, the contractor

needs to know how much he is going to bid at the various key events and the dates at which he is going to promise delivery of various portions of the project. If we are dealing with a deterministic network (in both duration and cost), there is no ambiguity as to either of these quantities. The completion date takes on a value that has probability 1, as does the cost. In this case, then, a bid can be prepared and submitted with no risk to the contractor, and hence, his profit is secure.

More realistically, however, one cannot assume that all quantities in the model are deterministic. We must assume that activity duration is only defined in a probabilistic sense as a random variable, and cost is some function of the duration (and subsequently a random variable itself), or is conditional on the observed duration. This results in the realization times and costs at key events being defined as random variables.

4.4.1 Realization Time Distribution

In a given project, a contractor may have many different key events at which he must deliver a product, or a portion of a product, to the owner of the project. The contractor, in his proposal, will give target dates by which he must deliver these products, and in the case of an overrun he will assume some penalty for being late. Since there is most likely competition for the project, a contractor must

submit a competitive delivery date for any chance of getting the contract. If this is the case, the contractor will be assuming a risk of observing realization times of the key events beyond those at which he promised delivery. To clarify this, the contractor's bid options include two extreme strategies: 1) Very conservative -- in this case, he will bid a long duration and correspondingly high cost which may well result in not receiving the contract and therefore making no money, and 2) Very optimistic, in which case he will bid an early completion time and corresponding low cost resulting in high probability of not fulfilling his promises, and consequently he will have heavy penalties and make no money either! Because of this environment, it is essential that he have accurate information on the distribution of realization time in order to properly assess his risk.

The problem of determining the pdf of the realization time of a node in a directed acyclic network has been one of the main areas of work in the area of activity networks since the very beginnings of the PERT model. Elmaghraby [6] discusses various methods of determining the pdf of a node (key event) in the PERT model, which include analytical approaches, approximations, bounding approaches, and Monte Carlo sampling. Once the pdf is determined by any one of these methods, the contractor can estimate the risk he is taking when giving a delivery date.

Notice that if we determine the realization time dis-

tributions of different key events in the network that have common activities, the resulting random variables are not independent. Because of this, at least in theory, when we are determining delivery dates for these dependent key events, we should consider them simultaneouly. In practice, however, when approximating the realization time distributions of these nodes, we normally assume independence of paths to each node. Under this assumption, the resulting realization time distributions are independent. Consequently, we can consider the delivery dates independently from each other.

4.4.2 Cost Distribution

Similarly, when a competitive bid is submitted, the contractor will assume a risk that the actual realization of the network cost is greater than his bid, thus incurring a loss. For this reason, it is essential also that a pdf of cost for the subgraph of a key event be determined. We now address the problem of determining the cost of an activity, and subsequently will examine the cost of a subgraph of a key event.

4.4.2.1 Activity Cost

As was mentioned previously, there are different costs that can be incurred due to an activity. In this thesis, we will consider only the following two costs: 1) Fixed cost

incurred at the inception of the activity, and 2) Variable cost incurred during the execution of the activity that is either a function of the activity duration or is a random variable whose pdf is conditional upon the activity duration. If the cost is a function of the duration (e.g., a linear function in the form of Cost = a + (b * duration)), then it is fairly straightforward to determine the pdf of the cost of the activity. But if the cost is a random variable that is conditional upon the actual activity duration, then to determine the cost pdf of the activity we must appeal to basic probability theory. Let C denote the cost of the activity, and Y denote its duration. Then we have the following relationship on the joint density of cost and duration, the conditional density of cost given duration, and the unconditional density on duration:

$$f_{C|Y}(c,y) = f(c,y)/f_Y(y) \tag{6}$$

To determine the unconditional cost density we must integrate the joint density with respect to duration. The following example will illustrate this procedure, along with the difficulties associated with it.

Let the duration be distributed exponentially with mean 10 ($f_Y(y) = 1/10 * \exp(-y/10)$), and cost, conditional on the activity duration, be distributed exponentially with mean 100*duration ($f_{C|Y}(c,y) = 1/100*y * \exp(-c/100*y)$). Then the unconditional distribution of cost is:

$$\int_0^{\infty} 1/1000y * \exp\{ -y/10-c/100*y \}\, dy \tag{7}$$

Clearly, this is not easy to evaluate. With many distributions, the unconditional distribution of cost could not be evaluated analytically.

4.4.2.2 Cost of Subgraphs

The cost of the subgraph of a key event is the sum of the costs of the individual activities of the subgraph. If we assume that the costs of activities are independent (although in actuality this may not necessarily be the case), then the distribution function of the cost of the subgraph will be the convolution of the individual activity cost distribution functions. This does not pose any theoretical problems, but an analytical solution for a given network poses definite computational problems. This is due to the fact that only in very rare cases will the pdf's of the costs of individual activities be amenable to mathematical manipulation (e.g., normal, in which case the convolution is easily determined since the sum of normal r.v.'s is itself normal). More likely is the case where we need to sum random variables that assume different pdf's. An analytical solution would involve multiple integration, which is a formidable computational problem.

To bypass this difficulty in analytically determining the cost pdf's, we can appeal to approximating the cost distribution by the following methods: 1)Central Limit Theorem, and 2) Monte Carlo Sampling.

By the Central Limit Theorem, we know that, subject to mild assumptions, as the number of activities in the subgraph of a key event goes to infinity, the cost distribution of the key event may be approximated by the normal distribution with mean = Σ_i mean(i) and variance = Σ_i var(i). Unfortunately, we know that the number of activities in the subgraph of a key event is oftentimes not very large, and hence the application of the Central Limit Theorem can result in erroneous probability statements (remember that determining the cost distributions of the key events is something we want to do as accurately as possible, especially in the context of project bidding where poor approximation of the cost distribution can lead to a large monetary loss).

If we are unwilling to apply the Central Limit Theorem, we can approximate the cost pdf by Monte Carlo sampling. This method is straightforward population sampling and has some distinct advantages. First, the individual cost densities need not be determined, which, as was stated earlier, can be a difficult task. The only knowledge of the cost density necessary is the form relative to the duration of the activity (e.g., a linear function of duration, a normal distribution about a constant times the duration, etc.). Another advantage of Monte Carlo sampling is that when approximating the cost distribution we can guarantee that our approximate cost distribution is within a certain tolerance level from the true distribution (see sample size discus-

sion). Monte Carlo sampling will be discussed further in section 5.

4.4.2.3 Dependency of Cost Random Variables

As discussed previously, in the cases of (i) either S(i) or S(j) is a subset of the other, and (ii) S(i) and S(j) are disjoint, the cost of activities are unambiguously assigned to their respective key events. But when S(i) and S(j) partially overlap (neither is a subset of the other), difficulties arise in allocating the cost of the common activities to the key events. That problem was resolved by partitioning the cost of the common activities. Unfortunately, the act of dividing the cost of a common activity and assigning percentages of the cost to different key events results in cost allocations that are not independent r.v.'s. To illustrate this point, we use the network in Figure 4 with all durations Gamma distributed with t=2, lambda=1. As was previously discussed, the cost of activity 'a' is allocated to the two key events 4 and 5. The resulting cost values are:

$$\text{node 4:}\quad C(4) = prop(4) * cost(a) + cost(c) \tag{8}$$

$$\text{node 5:}\quad C(5) = prop(5) * cost(a) + cost(b) + cost(d) + cost(e) \tag{9}$$

where prop(i) = proportion of the cost of activity 'a' assigned to node i,

C(i) = cost of key event i, and

cost(i) = cost of activity i.

It is then clear that the costs at nodes 4 and 5 are not independent because both r.v.'s include the cost of activity 'a'.

Because of this fact, when we have determined the cost pdf's of the key events under specific allocation strategies and are deciding what the bid should be at each key event, we should not consider the key events independently of each other. We can define the joint cumulative probability distribution by:

$$F(c1,\ldots,cn) = \Pr\{\ C(1) \leq c1,\ldots,C(n) \leq cn\ \} \tag{10}$$

The marginal distribution of the cost at a key event can then be shown [12] to be:

$$F(ci) = \Pr\ \{\ C(1) \leq \infty,\ldots,C(i) \leq ci,\ldots,C(n) \leq \infty\ \} \tag{11}$$

Given this marginal distribution, we can determine a bid at one key event independently from the costs at other key events. Theoretically, this is a straightforward application of elementary probability theory. The application of this theory, though, is difficult to carry out because of its computational complexity.

Fortunately, in practice, it would be consistent to assume that the cost r.v.'s can be treated as independent r.v.'s, as was done in the approximation of the realization times of the key events.

4.5 Determination of Cash Flow

When confronted with a potential project, the contractor must know if he will maintain liquidity throughout the project. To this end, we need to determine the expected cash "outflow" for the duration of the project. When activity duration and cost are defined as random variables, the problem of scheduling activities becomes very complex, and therefore determining the expected disbursements for the duration of the project is also complex. There are two methods to which we can resort in solving this problem, both of which require extensive computing effort. These methods are: 1) Monte Carlo simulation, and 2) analytical approach. Monte Carlo simulation is a straightforward method of determining the expected costs incurred in given intervals over many realizations of the network. This will be discussed in section 5.

The analytical approach to solving this problem, although conceptually fairly simple, is no trivial exercise. It is outlined here to display the difficulty associated in determining the exact expected cash "outflow".

Denote the start time of activity 'i' by ST_i, and let $E[C(m,i)]$ denote the expected cost incurred in period 'm' due to activity 'i'. Clearly, $E[C(m,i)]$ is dependent on three elements: the df of the cost of the activity as a function of its duration, the start time of the activity, and the probability that it is 'in progress' in period 'm'.

The total expected disbursement in period 'm' could then be the sum of the individual expected costs; the sum extending over all activities that may be in progress in period 'm'.

It is evident that the determination of $E[C(m,i)]$ is no minor feat, even when using the approximation method of Dodin [3] to determine the df of ST_i. Consequently, the analytical determination of the expected cash "outflow" was abandoned, and attention was diverted to the use of MCS methods.

5. SOLUTION PROCEDURE

The procedure adopted in solving the bidding problem is Monte Carlo simulation (MCS). The main reason for the adoption of this approach is to allow the calculation of the cash "outflow" discussed in the previous section. As Elmaghraby states [5], "whenever analytical approaches fail, or appear to overwhelm one's capacity to obtain numerical answers, one turns to population sampling and statistical techniques." Also, since the application of ANs to the bidding of a project for PERT networks has never been addressed before, MCS seems to be a logical first step in the solution of the problem.

The MCS technique generates a duration for each arc in the network from its distribution, and then calculates the realization times for the key events in the network. This is considered one sample. The result over many of these samples is a realization time distribution for each key event in the network. In addition, for the bidding problem, we must generate a cost for each activity according to its distribution (thus resulting in a cost distribution over many samples), as well as calculate the cash "outflow" in each period for each sample (resulting in an expected cash "outflow" over many samples), and calculate the cash "inflow" in each period for each sample (resulting in an expected cash "inflow" over many samples).

There are two theoretical concerns involved with MCS:

1) Size of the sample to guarantee a pre-assigned confidence, and 2) Use of variance-reduction techniques to increase precision (or minimize sample size). A discussion of sample size follows. The use of variance-reduction techniques is discussed by Elmaghraby [5] and [6], and will not be addressed in this thesis.

Sample Size

The main objective of the bidding problem is the determination of the distribution functions of realization times and costs of the key events in the network. If we use MCS to determine these df's, we have to make the sample size sufficient to estimate them with a given precision. This sample size can be determined by means of the Kolmogorov-Smirnov statistic as discussed by Elmaghraby [5] and tabulated by Hoel [9] and Massey [11] for limited values of sample size and confidence. For completeness, these concepts are discussed next.

If we assume the true distribution function, F_n, is continuous, we can make statements concerning the greatest absolute difference between the sample distribution function and the true distribution function. Let $G_{nK}(t)$ denote the sample df derived from a sample of size K, and $D_{nK}=\sup_t |G_{nK}(t)-F_n(t)|$. Then the probability that D_K is less than a specified $d/\sqrt{K}$ is asymptotically given by

$$\lim_{K\to\infty} \Pr\{ D_{nK} \le d/\sqrt{K} \} = \Sigma_{i=-\infty}^{\infty} (-1)^{i} * \exp(-2*i^2*d^2) \qquad (14)$$

The values of d for different confidence levels have been

tabulated by Smirnov [14] and the asymptotic probability of $\sqrt{K} * D_{nK} \leq d$ can then be obtained. As an example, suppose we wish to estimate F_n by G_{nK} such that the maximum deviation between the two functions does not exceed 0.01 in absolute value more than 5% of the time. For the asymptotic probability $\Pr\{\sqrt{K} * D_{nK} \leq d\} = .95$, we have $d=1.36$. Since $D_{nK} = .01$, we solve for $\sqrt{K} * 0.01 \geq 1.36$ to obtain $K \geq 18{,}496$.

Of course, in the bidding problem, we are concerned with estimating many df's simultaneously. This does not, however, introduce any complications even though some of the random variables whose df's we are estimating are not independent (specifically, cost and duration at key events). This is because, for each df we are estimating, the samples are independent and identically distributed. Therefore, we can estimate all of the df's simultaneously using the sample size determined above, and we can be confident that our Monte Carlo approximations to the distribution functions are accurate enough to make statements on the realization times and costs of key events.

6. COMPUTER PROGRAM

Software has been developed, written in Microsoft FORTRAN77, that allows the user to interactively determine a bid for a project that can be modeled by a directed acyclic network. The program listings are in the appendix, and an example will be presented section 6.4.

6.1 Program Explanation

The computer software developed is divided into three main programs: ANC-IN, AN-COST, and CASH, and an optional program DOD1TRNS.

Program ANC-IN prepares the data necessary for the Monte Carlo simulation. It also has the capability to randomly generate an AN if desired for research purposes. This program is a modification of the input program used by Dodin [3] to approximate the realization time distributions of nodes in a directed acyclic network. This provides the capability to use Dodin's program using the same data file, which allows comparison of the Monte Carlo-generated realization time distributions and the approximate distributions provided by Dodin.

Program DOD1TRNS, written by Dodin [3], calculates the means and standard deviations of the key events which are required by program AN-COST for use in the sharing of costs of common activities as well as in determining the endpoints of the empirical distributions. The means and standard de-

viations are then placed in a data file which is read by program AN-COST. If these values are available from another source, they can simply be placed in the data file and DOD1TRNS need not be run.

Program AN-COST actually performs the Monte Carlo simulation to determine the realization time distributions and cost distributions of the key events, the expected cash "inflow", and the expected cash "outflow". To do this, it determines the cost separation percentages of the common activities and calculates the required sample size before the simulation begins. The results of the run are sent to various data files for perusal.

Program CASH allows cash flow calculations to be performed, as well as allowing the user to view the results of the Monte Carlo simulation. It reads a data file created by AN-COST, and interactively assists the user in determining a bid package.

6.2 Program Assumptions

The following assumptions were made in implementing the computer program.

General:
1) Directed acyclic networks only.
2) The network has at most 100 activities
3) The network has at most 50 nodes (this is a limitation of DOD1TRNS(see 6, below), not of AN-COST).
4) Maximum project duration is 100 periods.
5) All activities begin at their earliest start times.
6) The means and std. dev.s of the key nodes are

known prior to running program AN-COST. (If these parameters are not known, they can be obtained by running program DOD1TRNS.)

Duration: --Activity duration is distributed as one of the following:

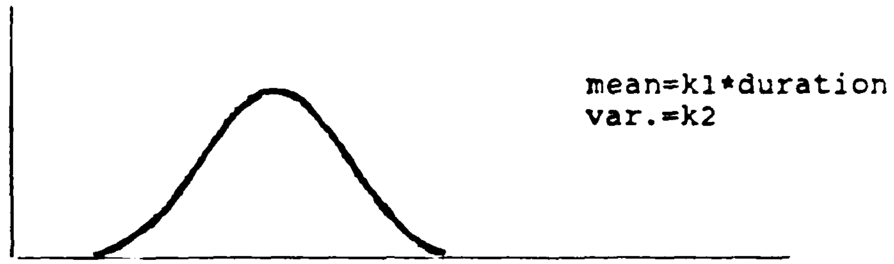
1) Uniform
2) Triangular
3) Normal
4) Exponential
5) Gamma
6) Beta
7) Discrete (user defined as ordered pairs)

Cost: Activity cost consists of a fixed cost assessed at the start of the activity, and a cost that is either a linear function of the duration or a cost that follows a density conditional on the realized duration. The following cost functions are available:

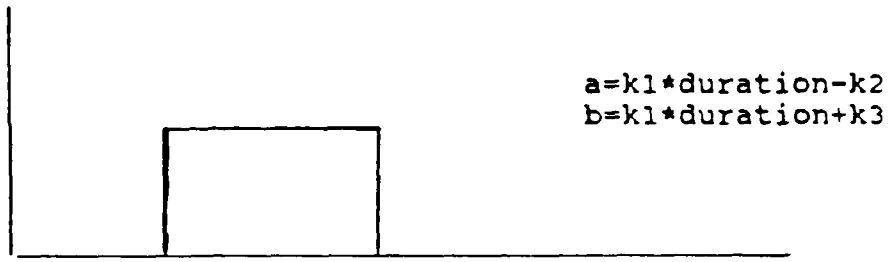
1) Constant * Duration
2) Normal (mean=constant*duration, var.=given) (see discussion, below)
3) Uniform (the endpoints of the uniform dist. are [constant*duration-given val., constant*duration+given val.]) (see discussion, below)
4) Triangular (mode=constant*duration, min=mode-given, max=mode+given) (see discussion, below)

Cost functions 2, 3, and 4 are illustrated in Figure 10. For a normal cost function, the mean is a user-specified constant times the activity duration, and the variance is a user-specified constant.

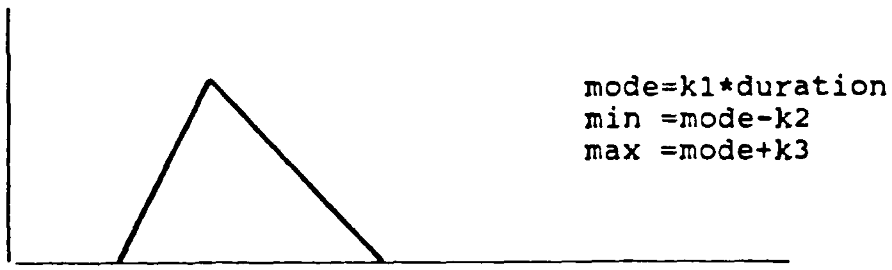
For a uniform cost function, the endpoints of the distribution are a user-specified constant, k1, times the activity duration minus another user-specified value, k2, and k1 times the activity duration plus another user-specified value, k3. Therefore, the distribution does not have to be symmetric about k1 * duration.



a) normal cost distribution



b) uniform cost distribution



c) triangular cost distribution

Figure 9: Illustration of Cost Functions

For a triangular cost function, the mode of the distribution is a user-specified value, k1, times the activity duration. The minimum is the mode minus another user-specified value, k2, and the maximum is the mode plus another user-specified value, k3. Again, the distribution is not necessarily symmetric.

6.3 Program Usage

To use the programs to make bid packages, the following steps must be taken:

1) Create a data file containing the duration and cost information of the network. This can be done using program ANC-IN. This program is an alteration of the input program for program DODIN1, which determines the PDF of realization time of nodes in a network. The data is put in the file in such a manner that DODIN1 can be run using the same data file.

2) Put the means and standard deviations of the key events in a data file named TRNS.ANC in increasing order of node number. If the means and std. dev.s are not known a priori, a run of program DOD1TRNS will determine them and place them in the data file. Program DOD1TRNS, which can be used to do all the calculations of DODIN1 with the exception of generation of a network and MCS, is a skeleton version of DODIN1.

3) Run program AN-COST. This program performs the MCS to determine realization time distributions and cost distributions of the key events of the network. This program also calculates the expected cash "inflow" and "outflow" for each period in the network. The cash flow data is sent to file CASHFLOW.DAT, which is used in program CASH to perform cash flow calculations.

4) Run program CASH if cash flow calculations are

desired. Program CASH will allow the user to view the data from the MCS, determine a FaRM in deterministic problems, and determine the expected profit in a probabilistic problem. It allows viewing of the distribution functions, and determines the percentiles of the distributions as an aid in making decisions on the bids at the key events.

6.4 Example

Consider the network previously used in Figure 1, reproduced in Figure 10 for convenience. For this example, let the activities durations be normally distributed and the costs be linear functions of the durations with the parameters listed in Table 8. Then running program ANC-IN will yield the input data file listed in Table 9.

Table 8. Activity Distribution Parameters for Example Problem

Activity	Duration		Cost	
	Mean	Std. Dev.	Cost/day	Fixed Cost
1	2	0.5	100	500
2	4	1.0	300	1000
3	10	2.0	250	1000
4	4	1.0	150	200
5	6	1.5	250	1000
6	5	1.5	150	200
7	7	1.0	100	200
8	7	1.0	75	200
9	0	0.0	0	0
10	9	2.0	40	50
11	8	1.5	40	25
12	5	2.0	40	50
13	4	1.0	75	100
14	0	0.0	0	0
15	2	0.5	200	100
16	6	1.5	100	300

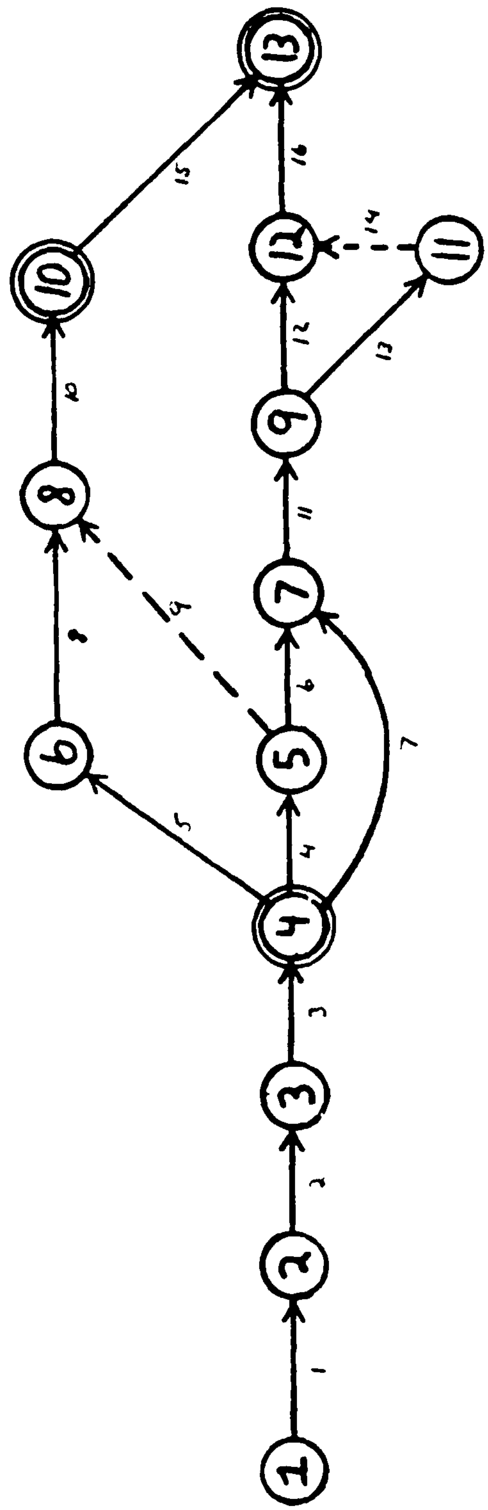


Figure 10: Network for Illustration of Computer Program

Table 9. Input Data File for Example Problem.*

```
      #nodes  #arcs                               #key events
.500     13      16      30     1     0     0          3        0

   key  events
   4    10    13

**start node    end node    duration dist.    #durations/
distribution parameters
   1      2      3      0
 .2000E+01  .5000E+00  .5000E+00  .3500E+01
   2      3      3      0
 .4000E+01  .1000E+01  .2000E+01  .6000E+01
   3      4      3      0
 .1000E+02  .2000E+01  .5000E+01  .1500E+02
   4      5      3      0
 .4000E+01  .1000E+01  .2000E+01  .6000E+01
   4      6      3      0
 .6000E+01  .1500E+01  .3000E+01  .9000E+01
   5      7      3      0
 .5000E+01  .1500E+01  .2000E+01  .8000E+01
   4      7      3      0
 .7000E+01  .1000E+01  .5000E+01  .9000E+01
   6      8      3      0
 .7000E+01  .1000E+01  .5000E+01  .9000E+01
   5      8      7      1
 .0000E+00  .1000E+01
   8     10      3      0
 .9000E+01  .2000E+01  .5000E+01  .1300E+02
   7      9      3      0
 .8000E+01  .1500E+01  .5000E+01  .1100E+02
   9     12      3      0
 .5000E+01  .2000E+01  .1000E+01  .9000E+01
   9     11      3      0
 .4000E+01  .1000E+01  .2000E+01  .6000E+01
  11     12      7      1
 .0000E+00  .1000E+01
  10     13      3      0
 .2000E+01  .5000E+00  .1000E+01  .3000E+01
  12     13      3      0
 .6000E+01  .1500E+01  .3000E+01  .9000E+01
```

*Numbers not labeled are used only by program DOD1TRNS in the determination of the mean and standard deviation of the key events (for more info. see Dodin [3]).

**See program listing for more details.

Table 9 (Continued)

*cost distribution/				
fixed cost	variable cost	distribution parameters		
1				
.5000E+03	.1000E+03	.0000E+00	.0000E+00	.0000E+00
1				
.1000E+04	.3000E+03	.0000E+00	.0000E+00	.0000E+00
1				
.1000E+04	.2500E+03	.0000E+00	.0000E+00	.0000E+00
1				
.2000E+03	.1500E+03	.0000E+00	.0000E+00	.0000E+00
1				
.1000E+04	.2500E+03	.0000E+00	.0000E+00	.0000E+00
1				
.2000E+03	.1500E+03	.0000E+00	.0000E+00	.0000E+00
1				
.2000E+03	.1000E+03	.0000E+00	.0000E+00	.0000E+00
1				
.2000E+03	.7500E+02	.0000E+00	.0000E+00	.0000E+00
1				
.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
1				
.5000E+02	.4000E+02	.0000E+00	.0000E+00	.0000E+00
1				
.2500E+02	.4000E+02	.0000E+00	.0000E+00	.0000E+00
1				
.5000E+02	.4000E+02	.0000E+00	.0000E+00	.0000E+00
1				
.1000E+03	.7500E+02	.0000E+00	.0000E+00	.0000E+00
1				
.0000E+00	.0000E+00	.0000E+00	.0000E+00	.0000E+00
1				
.1000E+03	.2000E+03	.0000E+00	.0000E+00	.0000E+00
1				
.3000E+03	.1000E+03	.0000E+00	.0000E+00	.0000E+00

*See program listing for more details.

If the means and standard deviations of the key events are not known, we next run program DOD1TRNS, which will place them in data file TRNS.ANC. Table 10 displays the file for this network.

Table 10: File TRNS.ANC for Example Problem

Key Event	Mean	Std. Dev.
4	.1600E+02	.2290E+01
10	.3800E+02	.3530E+01
13	.4400E+02	.6000E+01

The MCS, performed by program AN-COST, using a sample size of 18,496 (this sample size guarantees a tolerance of no more than .01 with a confidence of 95%) yields the output files shown in Tables 11, 12, 13, and 14. Table 11 contains the distribution information of duration of the key events as well as the criticality indices of the activities. Table 12 contains the distribution information of cost of the key events and total project cost. Table 13 displays a data file produced with some general information used by the simulation, and table 14 lists the data file CASHFLOW.DAT, produced by AN-COST for use by program CASH.

Table 11. Empirical Distributions of Durations and Activity Criticality Indices.

---- KEY EVENT 4 ------

Pr (X <= 10) =	.0000
Pr (10.000< X <= 10.550) =	.0006
Pr (10.550< X <= 11.100) =	.0019
Pr (11.100< X <= 11.650) =	.0042
Pr (11.650< X <= 12.200) =	.0098
Pr (12.200< X <= 12.750) =	.0167
Pr (12.750< X <= 13.300) =	.0305
Pr (13.300< X <= 13.850) =	.0481
Pr (13.850< X <= 14.400) =	.0735
Pr (14.400< X <= 14.950) =	.0917
Pr (14.950< X <= 15.500) =	.1140
Pr (15.500< X <= 16.050) =	.1169
Pr (16.050< X <= 16.600) =	.1224
Pr (16.600< X <= 17.150) =	.1105
Pr (17.150< X <= 17.700) =	.0902
Pr (17.700< X <= 18.250) =	.0670
Pr (18.250< X <= 18.800) =	.0444
Pr (18.800< X <= 19.350) =	.0268
Pr (19.350< X <= 19.900) =	.0149
Pr (19.900< X <= 20.450) =	.0092
Pr (20.450< X <= 21.000) =	.0044
Pr (X > 21) =	.0022

Pr (X <= 10.000) =	.0000
Pr (X <= 10.550) =	.0006
Pr (X <= 11.100) =	.0025
Pr (X <= 11.650) =	.0068
Pr (X <= 12.200) =	.0165
Pr (X <= 12.750) =	.0333
Pr (X <= 13.300) =	.0638
Pr (X <= 13.850) =	.1119
Pr (X <= 14.400) =	.1853
Pr (X <= 14.950) =	.2770
Pr (X <= 15.500) =	.3911
Pr (X <= 16.050) =	.5080
Pr (X <= 16.600) =	.6304
Pr (X <= 17.150) =	.7409
Pr (X <= 17.700) =	.8311
Pr (X <= 18.250) =	.8981
Pr (X <= 18.800) =	.9425
Pr (X <= 19.350) =	.9693
Pr (X <= 19.900) =	.9842
Pr (X <= 20.450) =	.9934
Pr (X <= 21.000) =	.9978

Table 11 (Continued)

THE MEAN OF THE DIST. =	16.0044
THE VARIANCE OF THE DIST. =	3.1521
THE STD. DEV. OF THE DIST. =	1.7754

Table 11 (Continued)

```
              ---- KEY EVENT 10  ------

   Pr { X <=   29 } =    .0000
   Pr {    29.000< X <=   29.850} =   .0009
   Pr {    29.850< X <=   30.700} =   .0020
   Pr {    30.700< X <=   31.550} =   .0045
   Pr {    31.550< X <=   32.400} =   .0106
   Pr {    32.400< X <=   33.250} =   .0203
   Pr {    33.250< X <=   34.100} =   .0362
   Pr {    34.100< X <=   34.950} =   .0527
   Pr {    34.950< X <=   35.800} =   .0831
   Pr {    35.800< X <=   36.650} =   .0955
   Pr {    36.650< X <=   37.500} =   .1159
   Pr {    37.500< X <=   38.350} =   .1269
   Pr {    38.350< X <=   39.200} =   .1213
   Pr {    39.200< X <=   40.050} =   .1027
   Pr {    40.050< X <=   40.900} =   .0865
   Pr {    40.900< X <=   41.750} =   .0602
   Pr {    41.750< X <=   42.600} =   .0377
   Pr {    42.600< X <=   43.450} =   .0203
   Pr {    43.450< X <=   44.300} =   .0122
   Pr {    44.300< X <=   45.150} =   .0061
   Pr {    45.150< X <=   46.000} =   .0025
   Pr { X >    46 } =    .0019

   Pr { X <=      29.000 } =     .0000
   Pr { X <=      29.850 } =     .0009
   Pr { X <=      30.700 } =     .0029
   Pr { X <=      31.550 } =     .0074
   Pr { X <=      32.400 } =     .0180
   Pr { X <=      33.250 } =     .0383
   Pr { X <=      34.100 } =     .0746
   Pr { X <=      34.950 } =     .1272
   Pr { X <=      35.800 } =     .2103
   Pr { X <=      36.650 } =     .3058
   Pr { X <=      37.500 } =     .4218
   Pr { X <=      38.350 } =     .5487
   Pr { X <=      39.200 } =     .6700
   Pr { X <=      40.050 } =     .7727
   Pr { X <=      40.900 } =     .8592
   Pr { X <=      41.750 } =     .9193
   Pr { X <=      42.600 } =     .9570
   Pr { X <=      43.450 } =     .9773
   Pr { X <=      44.300 } =     .9895
   Pr { X <=      45.150 } =     .9956
   Pr { X <=      46.000 } =     .9981
```

Table 11 (Continued)

THE MEAN OF THE DIST. =	38.0164
THE VARIANCE OF THE DIST. =	7.2432
THE STD. DEV. OF THE DIST. =	2.6913

Table 11 (Continued)

---- KEY EVENT 13 ------

Pr { X <= 35 } =	.0002
Pr { 35.000< X <= 35.850} =	.0007
Pr { 35.850< X <= 36.700} =	.0014
Pr { 36.700< X <= 37.550} =	.0044
Pr { 37.550< X <= 38.400} =	.0087
Pr { 38.400< X <= 39.250} =	.0167
Pr { 39.250< X <= 40.100} =	.0301
Pr { 40.100< X <= 40.950} =	.0465
Pr { 40.950< X <= 41.800} =	.0675
Pr { 41.800< X <= 42.650} =	.0866
Pr { 42.650< X <= 43.500} =	.1090
Pr { 43.500< X <= 44.350} =	.1165
Pr { 44.350< X <= 45.200} =	.1171
Pr { 45.200< X <= 46.050} =	.1043
Pr { 46.050< X <= 46.900} =	.0926
Pr { 46.900< X <= 47.750} =	.0680
Pr { 47.750< X <= 48.600} =	.0512
Pr { 48.600< X <= 49.450} =	.0338
Pr { 49.450< X <= 50.300} =	.0195
Pr { 50.300< X <= 51.150} =	.0128
Pr { 51.150< X <= 52.000} =	.0064
Pr { X > 52 } =	.0061

Pr { X <= 35.000 } =	.0002
Pr { X <= 35.850 } =	.0009
Pr { X <= 36.700 } =	.0023
Pr { X <= 37.550 } =	.0067
Pr { X <= 38.400 } =	.0154
Pr { X <= 39.250 } =	.0321
Pr { X <= 40.100 } =	.0622
Pr { X <= 40.950 } =	.1087
Pr { X <= 41.800 } =	.1761
Pr { X <= 42.650 } =	.2627
Pr { X <= 43.500 } =	.3717
Pr { X <= 44.350 } =	.4882
Pr { X <= 45.200 } =	.6053
Pr { X <= 46.050 } =	.7096
Pr { X <= 46.900 } =	.8022
Pr { X <= 47.750 } =	.8702
Pr { X <= 48.600 } =	.9214
Pr { X <= 49.450 } =	.9552
Pr { X <= 50.300 } =	.9748
Pr { X <= 51.150 } =	.9875
Pr { X <= 52.000 } =	.9939

Table 11 (Continued)

THE MEAN OF THE DIST. =	44.4840
THE VARIANCE OF THE DIST. =	8.3119
THE STD. DEV. OF THE DIST. =	2.8830

Table 12. Empirical Distributions of Costs

---- PROJECT COST ----

Pr { X <=13273 } =	.0003
Pr {13273.000< X <=13459.900} =	.0032
Pr {13459.900< X <=13646.800} =	.0050
Pr {13646.800< X <=13833.700} =	.0094
Pr {13833.700< X <=14020.600} =	.0192
Pr {14020.600< X <=14207.500} =	.0297
Pr {14207.500< X <=14394.400} =	.0487
Pr {14394.400< X <=14581.300} =	.0667
Pr {14581.300< X <=14768.200} =	.0930
Pr {14768.200< X <=14955.100} =	.1055
Pr {14955.100< X <=15142.000} =	.1170
Pr {15142.000< X <=15328.900} =	.1174
Pr {15328.900< X <=15515.800} =	.1066
Pr {15515.800< X <=15702.700} =	.0922
Pr {15702.700< X <=15889.600} =	.0663
Pr {15889.600< X <=16076.500} =	.0522
Pr {16076.500< X <=16263.400} =	.0317
Pr {16263.400< X <=16450.300} =	.0174
Pr {16450.300< X <=16637.200} =	.0104
Pr {16637.200< X <=16824.100} =	.0048
Pr {16824.100< X <=17011.000} =	.0018
Pr { X > 17011 } =	.0015

Pr { X <= 13273.000 } =	.0003
Pr { X <= 13459.900 } =	.0035
Pr { X <= 13646.800 } =	.0085
Pr { X <= 13833.700 } =	.0179
Pr { X <= 14020.600 } =	.0371
Pr { X <= 14207.500 } =	.0668
Pr { X <= 14394.400 } =	.1155
Pr { X <= 14581.300 } =	.1821
Pr { X <= 14768.200 } =	.2751
Pr { X <= 14955.100 } =	.3806
Pr { X <= 15142.000 } =	.4976
Pr { X <= 15328.900 } =	.6150
Pr { X <= 15515.800 } =	.7216
Pr { X <= 15702.700 } =	.8139
Pr { X <= 15889.600 } =	.8802
Pr { X <= 16076.500 } =	.9324
Pr { X <= 16263.400 } =	.9641
Pr { X <= 16450.300 } =	.9815
Pr { X <= 16637.200 } =	.9919
Pr { X <= 16824.100 } =	.9966
Pr { X <= 17011.000 } =	.9985

Table 12 (Continued)

THE MEAN OF THE DIST. = 15145.6822

THE VARIANCE OF THE DIST. = 390349.5940

THE STD. DEV. OF THE DIST. = 624.7796

Table 12 (Continued)

```
                ---- KEY EVENT  4 ------

Pr { X <= 5070 } =    .0004
Pr { 5070.000< X <= 5203.800} =   .0024
Pr { 5203.800< X <= 5337.600} =   .0048
Pr { 5337.600< X <= 5471.400} =   .0097
Pr { 5471.400< X <= 5605.200} =   .0192
Pr { 5605.200< X <= 5739.000} =   .0298
Pr { 5739.000< X <= 5872.800} =   .0504
Pr { 5872.800< X <= 6006.600} =   .0679
Pr { 6006.600< X <= 6140.400} =   .0922
Pr { 6140.400< X <= 6274.200} =   .1080
Pr { 6274.200< X <= 6408.000} =   .1163
Pr { 6408.000< X <= 6541.800} =   .1181
Pr { 6541.800< X <= 6675.600} =   .1069
Pr { 6675.600< X <= 6809.400} =   .0914
Pr { 6809.400< X <= 6943.200} =   .0684
Pr { 6943.200< X <= 7077.000} =   .0475
Pr { 7077.000< X <= 7210.800} =   .0293
Pr { 7210.800< X <= 7344.600} =   .0187
Pr { 7344.600< X <= 7478.400} =   .0103
Pr { 7478.400< X <= 7612.200} =   .0049
Pr { 7612.200< X <= 7746.000} =   .0020
Pr { X >  7746 } =    .0014

Pr { X <=  5070.000 } =    .0004
Pr { X <=  5203.800 } =    .0028
Pr { X <=  5337.600 } =    .0075
Pr { X <=  5471.400 } =    .0172
Pr { X <=  5605.200 } =    .0364
Pr { X <=  5739.000 } =    .0663
Pr { X <=  5872.800 } =    .1167
Pr { X <=  6006.600 } =    .1846
Pr { X <=  6140.400 } =    .2768
Pr { X <=  6274.200 } =    .3848
Pr { X <=  6408.000 } =    .5011
Pr { X <=  6541.800 } =    .6192
Pr { X <=  6675.600 } =    .7262
Pr { X <=  6809.400 } =    .8176
Pr { X <=  6943.200 } =    .8860
Pr { X <=  7077.000 } =    .9335
Pr { X <=  7210.800 } =    .9628
Pr { X <=  7344.600 } =    .9815
Pr { X <=  7478.400 } =    .9918
Pr { X <=  7612.200 } =    .9966
Pr { X <=  7746.000 } =    .9986
```

Table 12 (Continued)

THE MEAN OF THE DIST. =	6407.0008
THE VARIANCE OF THE DIST. =	198487.0996
THE STD. DEV. OF THE DIST. =	445.5189

Table 12 (Continued)

---- KEY EVENT 10 ------

Pr { X <= 3448 } =	.0001
Pr { 3448.000< X <= 3548.800} =	.0011
Pr { 3548.800< X <= 3649.600} =	.0037
Pr { 3649.600< X <= 3750.400} =	.0094
Pr { 3750.400< X <= 3851.200} =	.0187
Pr { 3851.200< X <= 3952.000} =	.0325
Pr { 3952.000< X <= 4052.800} =	.0497
Pr { 4052.800< X <= 4153.600} =	.0704
Pr { 4153.600< X <= 4254.400} =	.0925
Pr { 4254.400< X <= 4355.200} =	.1031
Pr { 4355.200< X <= 4456.000} =	.1180
Pr { 4456.000< X <= 4556.800} =	.1114
Pr { 4556.800< X <= 4657.600} =	.1076
Pr { 4657.600< X <= 4758.400} =	.0902
Pr { 4758.400< X <= 4859.200} =	.0712
Pr { 4859.200< X <= 4960.000} =	.0526
Pr { 4960.000< X <= 5060.800} =	.0323
Pr { 5060.800< X <= 5161.600} =	.0201
Pr { 5161.600< X <= 5262.400} =	.0091
Pr { 5262.400< X <= 5363.200} =	.0043
Pr { 5363.200< X <= 5464.000} =	.0015
Pr { X > 5464 } =	.0003

Pr { X <= 3448.000 } =	.0001
Pr { X <= 3548.800 } =	.0012
Pr { X <= 3649.600 } =	.0049
Pr { X <= 3750.400 } =	.0143
Pr { X <= 3851.200 } =	.0330
Pr { X <= 3952.000 } =	.0655
Pr { X <= 4052.800 } =	.1152
Pr { X <= 4153.600 } =	.1856
Pr { X <= 4254.400 } =	.2781
Pr { X <= 4355.200 } =	.3812
Pr { X <= 4456.000 } =	.4992
Pr { X <= 4556.800 } =	.6107
Pr { X <= 4657.600 } =	.7183
Pr { X <= 4758.400 } =	.8085
Pr { X <= 4859.200 } =	.8796
Pr { X <= 4960.000 } =	.9323
Pr { X <= 5060.800 } =	.9646
Pr { X <= 5161.600 } =	.9847
Pr { X <= 5262.400 } =	.9938
Pr { X <= 5363.200 } =	.9982
Pr { X <= 5464.000 } =	.9997

Table 12 (Continued)

THE MEAN OF THE DIST. = 4458.9100

THE VARIANCE OF THE DIST. = 110690.6299

THE STD. DEV. OF THE DIST. = 332.7020

Table 12 (Continued)

```
        ---- KEY EVENT 13  ------

Pr { X <= 3464 } =    .0002
Pr { 3464.000< X <= 3545.300} =   .0028
Pr { 3545.300< X <= 3626.600} =   .0039
Pr { 3626.600< X <= 3707.900} =   .0095
Pr { 3707.900< X <= 3789.200} =   .0191
Pr { 3789.200< X <= 3870.500} =   .0309
Pr { 3870.500< X <= 3951.800} =   .0486
Pr { 3951.800< X <= 4033.100} =   .0700
Pr { 4033.100< X <= 4114.400} =   .0882
Pr { 4114.400< X <= 4195.700} =   .1038
Pr { 4195.700< X <= 4277.000} =   .1167
Pr { 4277.000< X <= 4358.300} =   .1196
Pr { 4358.300< X <= 4439.600} =   .1058
Pr { 4439.600< X <= 4520.900} =   .0897
Pr { 4520.900< X <= 4602.200} =   .0723
Pr { 4602.200< X <= 4683.500} =   .0514
Pr { 4683.500< X <= 4764.800} =   .0316
Pr { 4764.800< X <= 4846.100} =   .0187
Pr { 4846.100< X <= 4927.400} =   .0108
Pr { 4927.400< X <= 5008.700} =   .0044
Pr { 5008.700< X <= 5090.000} =   .0017
Pr { X >  5090 } =    .0005

Pr { X <=  3464.000 } =    .0002
Pr { X <=  3545.300 } =    .0030
Pr { X <=  3626.600 } =    .0070
Pr { X <=  3707.900 } =    .0164
Pr { X <=  3789.200 } =    .0356
Pr { X <=  3870.500 } =    .0664
Pr { X <=  3951.800 } =    .1150
Pr { X <=  4033.100 } =    .1850
Pr { X <=  4114.400 } =    .2732
Pr { X <=  4195.700 } =    .3770
Pr { X <=  4277.000 } =    .4937
Pr { X <=  4358.300 } =    .6133
Pr { X <=  4439.600 } =    .7191
Pr { X <=  4520.900 } =    .8088
Pr { X <=  4602.200 } =    .8811
Pr { X <=  4683.500 } =    .9324
Pr { X <=  4764.800 } =    .9640
Pr { X <=  4846.100 } =    .9826
Pr { X <=  4927.400 } =    .9935
Pr { X <=  5008.700 } =    .9978
Pr { X <=  5090.000 } =    .9995
```

Table 12 (Continued)

THE MEAN OF THE DIST. = 4279.7714

THE VARIANCE OF THE DIST. = 73074.0281

THE STD. DEV. OF THE DIST. = 270.3221

Table 13. General Output File for Example Problem

THE PERIOD LENGTH = 5

THE PERC. REQUIRED AT START = .000

THE INFLATION RATE = .10000

THE INITIAL RANDOM NUMBER SEED = 123454.00

VALUES OF ENDPOINTS FOR EMPIRICAL DIST.'S ARE:

	LEFT-END	RIGHT-END	INT.-WIDTH
KEY EVENT 4			
REAL. TIME DIST.	10	21	.55
COST FUNC.	5070	7746	133.80
KEY EVENT 10			
REAL. TIME DIST.	29	46	.85
COST FUNC.	3448	5464	100.80
KEY EVENT 13			
REAL. TIME DIST.	35	52	.85
COST FUNC.	3464	5090	81.30
TOTAL PROJECT COST	13273	17011	186.90

Table 14. File CASHFLOW.DAT for Example Problem.

```
period length
     5
max. periods
    14
expected cash outflow per period
 .277648E+04 .212517E+04 .173647E+04 .325721E+04 .197417E+04
 .746016E+03 .746758E+03 .107538E+04 .606587E+03 .973920E+02
 .374120E+01 .952881E-02 .000000E+00 .000000E+00
#key events
    3
key events
    4   10   13
expected cash inflow for key event 4
 .000000E+00 .132299E+01 .169048E+04 .461520E+04 .999844E+02
 .000000E+00 .000000E+00 .000000E+00 .000000E+00 .000000E+00
 .000000E+00 .000000E+00 .000000E+00 .000000E+00
expected cash inflow for key event 10
 .000000E+00 .000000E+00 .000000E+00 .000000E+00 .000000E+00
 .454279E+01 .542953E+03 .281885E+04 .106864E+04 .239204E+02
 .000000E+00 .000000E+00 .000000E+00 .000000E+00
expected cash inflow for key event 13
 .000000E+00 .000000E+00 .000000E+00 .000000E+00 .000000E+00
 .000000E+00 .116708E+01 .229862E+03 .219535E+04 .170776E+04
 .144858E+03 .772703E+00 .000000E+00 .000000E+00

%received at proj.start    money received at proj. start
     .000000E+00                .000000E+00

    mean       std.dev.    (duration/cost of key events)
 .160044E+02 .177541E+01
 .640700E+04 .445518E+03
 .380164E+02 .269132E+01
 .445891E+04 .332702E+03
 .444840E+02 .288304E+01
 .427977E+04 .270322E+03
    mean       std.dev.    (total proj. cost)
 .151456E+05 .624779E+03

empirical distribution functions

   lower endpt.  upper endpt.  #cells  cell width
        10            21         20      .55000
 .000000E+00 .594723E-03 .194636E-02 .421712E-02 .978590E-02
 .167063E-01 .305471E-01 .480644E-01 .734753E-01 .916955E-01
 .114024E+00 .116944E+00 .122404E+00 .110456E+00 .902357E-01
 .670415E-01 .443879E-01 .268166E-01 .148680E-01 .919117E-02
 .437932E-02 .221669E-02
      5070         7746         20      133.80000
 .378460E-03 .237889E-02 .475778E-02 .973183E-02 .191933E-01
 .298442E-01 .503892E-01 .679065E-01 .921820E-01 .108023E+00
```

Table 14 (Continued)

.116349E+00	.118079E+00	.106942E+00	.914251E-01	.684472E-01
.474697E-01	.293036E-01	.187067E-01	.102724E-01	.486591E-02
.200043E-02	.135164E-02			
29	46	20	.85000	
.000000E+00	.919117E-03	.200043E-02	.448745E-02	.105968E-02
.203287E-01	.362240E-01	.526600E-01	.830990E-01	.955341E-01
.115917E+00	.126892E+00	.121323E+00	.102670E+00	.865051E-01
.601751E-01	.376838E-01	.202746E-01	.122188E-01	.605536E-02
.254109E-02	.189230E-02			
3448	5464	20	100.80000	
.540657E-04	.113538E-02	.367647E-02	.940743E-02	.187067E-01
.324935E-01	.497404E-01	.703936E-01	.925064E-01	.103103E+00
.118025E+00	.111429E+00	.107590E+00	.902357E-01	.711505E-01
.526059E-01	.323313E-01	.201124E-01	.913711E-02	.432525E-02
.151384E-02	.324394E-03			
35	52	20	.85000	
.162197E-03	.702854E-03	.140570E-02	.443339E-02	.870458E-02
.166522E-01	.301146E-01	.464965E-01	.674740E-01	.865592E-01
.108996E+00	.116511E+00	.117052E+00	.104346E+00	.926146E-01
.679606E-01	.512002E-01	.338451E-01	.195177E-01	.127595E-01
.643382E-02	.605536E-02			
3464	5090	20	81.30000	
.216263E-03	.281141E-02	.394679E-02	.946150E-02	.191392E-01
.308715E-01	.485510E-01	.700151E-01	.882353E-01	.103752E+00
.116673E+00	.119647E+00	.105752E+00	.896950E-01	.722859E-01
.513624E-01	.315743E-01	.186526E-01	.108131E-01	.437932E-02
.167603E-02	.486591E-03			
13273	17011	20	186.90000	
.324394E-03	.318987E-02	.497404E-02	.940743E-02	.191933E-01
.297361E-01	.486591E-01	.666630E-01	.929930E-01	.105482E+00
.116998E+00	.117376E+00	.106617E+00	.922361E-01	.663386E-01
.521734E-01	.317365E-01	.174091E-01	.103806E-01	.475778E-02

<u>sample corr. coef.</u> (ρ) <u>least squares estimators</u> ($C=\beta_0+\beta_1 Y+\epsilon$)

<u>cost vs. duration -- key event 4</u>

.972125E+00 .250284E+04 .243942E+03

<u>cost vs. duration -- key event 10</u>

.531348E+00 .196178E+04 .656853E+02

<u>cost vs. duration -- key event 13</u>

.543696E+00 .201204E+04 .509784E+02

<u>total project cost vs. project duration</u>

.750275E+00 .791297E+04 .162591E+03

<u>fraction of total project cost received at key events</u>

<u>key event 4</u>	<u>key event 10</u>	<u>key event 13</u>
.422863E+00	.294404E+00	.282732E+00

Program CASH is then run to determine an expected profit when given a particular bid strategy. Table 15 shows an output file from program CASH displaying a possible bid and its associated expected profit.

Table 15. Output File from Program CASH for Example [illegible]

Period	Interest*	Cash Balance**
I = 1	IBA = 4.363	CASH = [illegible]
I = 2	IBA = 8.444	CASH = -4414.4[illegible]
I = 3	IBA = 11.78[illegible]	CASH = -61[illegible]
I = 4	IBA = 14.61[illegible]	CASH = [illegible]
I = 5	IBA = 9.05[illegible]	CASH = [illegible]
I = 6	IBA = 10.299	CASH = -5184.6[illegible]
I = 7	IBA = 11.741	CASH = -61[illegible]
I = 8	IBA = 12.719	CASH = -664[illegible].8[illegible]
I = 9	IBA = 7.71[illegible]	CASH = -4[illegible]
I = 10	IBA = 1.288	CASH = -6[illegible]
I = 11	IAF = 1.509	CASH = 1156.36[illegible]
I = 12	IAF = 1.711	CASH = 1[illegible]11.[illegible]
I = 13	IAF = 1.715	CASH = 1[illegible]14.1[illegible]
I = 14	IAF = 1.717	CASH = [illegible].8[illegible]

TERMINAL CASH POSITION - 3525.580

*IBA -- interest on money borrowed,
IAF -- interest on money deposited.

**Cash balance at end of period.

```
---------BID PACKAGE----------

0.00 RECEIVED AT PROJECT START

KEY EVENT    4
     BID LEVEL     =   7715.31
     DELIVERY DATE =     21.00
     LATE PENALTY  =       .00
```

Table 15 (Continued)

```
KEY EVENT    10
     BID LEVEL        =    5369.42
     DELIVERY DATE    =      46.00
     LATE PENALTY     =        .00

KEY EVENT    11
     BID LEVEL        =    5153.[illegible]
     DELIVERY DATE    =      4[illegible].0[illegible]
     LATE PENALTY     =      1[illegible]

EXPECTED PROFIT =           .2[illegible]

EXPECTED PROJECT COST =          [illegible]

UNDER THE CONSTRAINTS:

INITIAL CAPITAL =       50[illegible].00

INTEREST RATE ON MONEY DEPOSIT = .1[illegible] %

INTEREST RATE ON MONEY BORROWED = .1[illegible] %

RETENTION RATE = .1200%
```

6.5 Discussion of Program

A question often raised about problem solutions is the efficiency of the procedure. Obviously, a Monte Carlo simulation, for a network of practical size, will have a long run time. Yet, considering that the program will be run only once for a particular project, the cost incurred will most likely be outweighed by the information obtained from the run.

The example problem (11 nodes, 16 activities, and a sample size of 18,496) ran for approximately 270 minutes on an AT&T PC 6300. This was slightly more than 1 sample per second. Obviously, a more powerful computer would reduce run time considerably.

Programs AN[illegible]-IN, AN-COST, and CASH have storage requirements of 94,264, 124,816, and 9[illegible],196 bytes, respectively, and were programmed on a PC with 640K RAM.

7. CONCLUSION AND DIRECTION FOR FUTURE RESEARCH

In the arena of project bidding, it is essential to submit a bid that will be competitive yet gainful to the bidder. For this reason, it is important that the manager have accurate and complete information on the project on which he is bidding. The computer package developed in this work gives the manager the distributions of duration and cost at the key events of the project, as well as the expected cash flow arising from the project execution. It also will assist him in cash flow calculations, and in the preparation of a bid package.

Application of AN theory to project bidding is an area of research that has been mostly ignored to this date. This work represents only the beginning of research in the area and needs empirical evidence to validate many of the assumptions of the model. A logical next step would be to approach the problem via analytical approximation as opposed to approximation by Monte Carlo simulation.

Further down the road, once the question of the determination of a bid has been satisfactorily investigated, is the problem of investment of additional resources in the project to yield earlier delivery dates for one or more of the key events of the project. This may be necessary in order to obtain the contract in a competitive environment if the delivery date is the overriding consideration. This problem goes deeper than the problem of optimal time-cost

trade-offs in directed acyclic networks because of the cash flow problem when determining a bid. The time-cost trade-off problem has not, to this date, been satisfactorily addressed in the case of probabilistic ANs.

Finally, a complete model of bidding should be cast in a game-theoretic framework, which has not been addressed in this thesis (here, we assumed a one-against-all model). This would take into account many factors such as the number of competitors bidding for the project, potential profit and loss functions, as well as multiple projects. The end result would be an overall bid strategy for the entire scope of many projects.

8. REFERENCES

1. Badger, Albert A. (1974). "Interrelating Project Estimates to CPM Schedules." AACE Transactions of the 18th National Meeting, 18, 38-42.

2. Case, Kenneth E. (1972). "On the Consideration of Variability in Cost Estimating." IEEE Transactions on Engineering Management, 19, No. 4, 114-118.

3. Dodin, Bajis M. (1982). "On the Completion Time of Stochastic PERT Networks." Unpublished Ph.D. Dissertation, N.C. State University.

4. Dodin, Bajis M., and Salah E. Elmaghraby (1985). "Approximating the Criticality Indices of the Activities in PERT Networks." Management Science, 31, No. 2, 207-223.

5. Elmaghraby, Salah E. (1977). Activity Networks: Project Planning and Control by Network Models, John Wiley & Sons, New York.

6. Elmaghraby, Salah E. (1986). "The Estimation of Some Network Parameters in the PERT Model of Activity Networks: Review and Critique", OR Report No. 207, North Carolina State Univ., Raleigh, NC.

7. Elmaghraby, Salah E. (1986). "Optimal Bidding Strategies in Project Activities", Research Proposal, North Carolina State University, Raleigh, NC.

8. Farid, Foad, and L. T. Boyer. (1985). "Fair and Reasonable Markup (FaRM) Pricing Model." Journal of Construction Engineering and Management, 111, No. 4, 374-390.

9. Hoel, Paul G. (1954). Introduction to Mathematical Statistics, third edition, John Wiley & Sons, New York.

10. Kleindorfer, G.B. (1971). "Bounding distributions for a stochastic acyclic network," Operations Research, 19, 1596-1601.

11. Massey, Frank, J, Jr. (1951). "The Kolmogorov-Smirnov Test for Goodness of Fit." Journal of the American Statistical Association, 46, 68-78.

12. Ross, Sheldon (1984). A First Course in Probability, Macmillan Publishing Company, New York.

13. Russell, Robert A. (1986). "A Comparison of Heuristics for Scheduling Projects With Cash Flows and Resource Restrictions," Management Science, 32, 1291-1300.

14. Smirnov, N. (1948). "Table for Estimating the Goodness of Fit of Empirical Distributions," Annals of Mathematical Statistics, 19, 279-281.

9. Appendix

Appendix 9.1. Program Listing of ANC-IN

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                          C
C     P R O G R A M     A N C - I N                       31  NOV  86      C
C                                                                          C
C     MODIFIED  BY  RUSSELL  S.  VOGTMANN                                  C
C                                                                          C
C          THIS  PROGRAM  PREPARES  AN  INPUT  DATA  FILE  FOR  PROGRAM  AN- C
C     COST.  IT  HAS  THE  CAPABILITY  TO  GENERATE  A  NETWORK  AT  RANDOM  IF C
C     DESIRED.                                                             C
C          IT  SHOULD  BE  SELF-EXPLANATORY.                               C
C                                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      COMMON   /PAR1/N,M,NRR,NCONT,MCS,NSIM,SCAL,NR(100),NT(50)
      COMMON   /PAR2/NS(100),NE(100),NDSTT(100),R(100,30),PR(100,30)
      COMMON   /PAR3/T(50,30),PT(50,30)
      COMMON   /PAR4/EX(100),STDX(100),VMIN(100),VMAX(100)
      DIMENSION    KEYN(50),NULT(100),NDS(100),AA(100),BB(100),CC(100)
      DIMENSION    DD(100),FC(100),DM(100)
      DIMENSION    NRC(100),P1(100)
      CHARACTER    FNAME*6,CFNAME*6
C
C     PRINT   PROGRAM   INTRODUCTION
C
      CALL   INFORM
C
C     THIS   PROGRAM   AREA   OBTAINS   GENERAL   INFORMATION   FROM   USER
C
      WRITE(*,301)
      READ(*,302)     FNAME
      OPEN(3,FILE=FNAME,STATUS='NEW')
      CALL   REED2(NANG)
      CALL   REED3(NCONT)
      MCS=0
      NSIM=0
      CALL   REED5(N,M)
      CALL   REED6(SCAL)
      CALL   REED7(KEY,KEYN,NRR,N)
      CALL   HALT
C
C     GENERATE   NETWORK   IF   DESIRED
C
      IF  (NANG.EQ.1)     CALL   GNRAT
      NNANG=0
      WRITE(3,202)      SCAL,N,M,NRR,NCONT,MCS,NSIM,KEY,NNANG
      WRITE(3,203)      (KEYN(I),I=1,KEY)
      IF(NANG.EQ.1)      THEN
C
C  **************************************************************
C     OBTAIN   DURATION   INFO  IF  NETWORK   HAS   BEEN   GENERATED.
C  **************************************************************
```

```
C
C
          CALL   REED8(NOI)
          DO  6  I=1,NOI
 6        CALL   REED9(I,NULT(I),NDS(I),NT(I))
C
            DO  7  I=1,NOI
            IF(NDS(I).NE.7)       THEN
              CALL   REED10(I,NDS(I),AA(I),BB(I),CC(I),DD(I))
            ELSE
              DO  12  J=1,NT(I)
 12           CALL   REED11(I,J,T(I,J),PT(I,J))
            ENDIF
 7          CONTINUE
C
         K=0
         DO  8  I=1,NOI
           IF(I.GT.1)     K=NULT(I-1)
         DO  9  II=K+1,NULT(I)
          WRITE(3,203)      NS(II),NE(II),NDS(I),NT(I)
          IF(NDS(I).NE.7)       THEN
            WRITE(3,205)AA(I),BB(I),CC(I),DD(I)
          ELSE
            WRITE(3,205)(T(I,J),PT(I,J),J=1,NT(I))
          ENDIF
  9      CONTINUE
  8      CONTINUE
C
C ******************************************************************
C             OBTAIN   DURATION    INFO.   IF  NETWORK   IS  NOT   GENERATED
C ******************************************************************
C
       ELSE
          DO  22  I=1,M
          CALL   REED12(I,NS(I),NE(I),NDSTT(I),NR(I))
          IF(NDSTT(I).NE.7)       THEN
            CALL   REED14(I,NDSTT(I),EX(I),STDX(I),VMIN(I),VMAX(I))
          ELSE
            DO  11  J=1,NR(I)
 11         CALL   REED15(I,J,R(I,J),PR(I,J))
          ENDIF
 22       CONTINUE
          DO  102  I=1,M
          WRITE(3,203)      NS(I),NE(I),NDSTT(I),NR(I)
          IF(NDSTT(I).NE.7)       THEN
            WRITE(3,205)EX(I),STDX(I),VMIN(I),VMAX(I)
          ELSE
            WRITE(3,205)(R(I,J),PR(I,J),J=1,NR(I))
          ENDIF
          CONTINUE
       [illegible]
        [illegible]
        [illegible]      PROGRAM   WILL   ASK   YOU   FOR   DATA   TO   RUN   ',/,
```

```
     &5X,'AN-COST.     IT DOES SOME PRELIMINARY   CHECKS  AND  PUTS  THE  ',/,
     &5X,'DATA   INTO  A  FILE.',//,
     &5X,'GIVE   A  NAME  FOR  THAT  FILE',/,
     &5X,'THE   NAME  SHOULD  BE  OF  FOUR  CHARACTERS   AND  THERE  SHOULD',/,
     &5X,'BE   TWO  ADDITIONAL   CHARACTERS   AT  THE  BEGINNING   ',/,
     &5X,'SPECIFYING     THE  DRIVE.',/,
     &5X,'AN   EXAMPLE   IS  B:PRES',/,
     &5X,'HOPE   THE  FILE  DOES  NOT  ALREADY   EXIST')
 201  FORMAT(I1)
 202  FORMAT(F5.3,10I5)
 203  FORMAT(14I5)
 205  FORMAT(6E11.4)
C
C ******************************************************************
C    PROGRAM   AREA  TO  OBTAIN   DATA  ON  ACTIVITY   COST
C ******************************************************************
C
      WRITE(*,303)
 303  FORMAT(5(/),5X,'NOW      YOU  WILL  BE  ASKED  FOR  INFORMATION    ON  THE
     &5X,'COST    DISTRIBUTIONS     OF  THE  ACTIVITIES     IN  THE  NETWORK
      IF  (NANG.NE.1)    THEN
C
C ******************************************************************
C        CASE   WHEN  ACTIVITY   NETWORK   IS  NOT  GENERATED   BY  PROGRAM
C ******************************************************************
C
         DO  33  I=1,M
           CALL   REED16 [illegible]
           CALL   REED17 [illegible]
 33      CONTINUE
C
[illegible]
```

```
          WRITE(3,205)      FC(I),DM(I),P1(I),VMIN(I),VMAX(I)
99        CONTINUE
98        CONTINUE
C
       ENDIF
C
       CLOSE(3,STATUS='KEEP')
       STOP
       END
C
C  ******************************************************************
C
       SUBROUTINE     INFORM
C
C  ******************************************************************
C
       WRITE(*,1)
       PAUSE
       WRITE(*,2)
       PAUSE
       WRITE(*,4)
       PAUSE
1      FORMAT(5X,'THIS      PROGRAM    PREPARES    THE   INPUT    FILES    FOR   PROGRAM',/,
     &5X,'AN COST      WHICH    DETERMINES     BID   PRICES    FOR   PROJECTS',/,
     &5X,'THAT     CAN   BE   MODELED     WITH   DIRECTED     ACYCLIC     NETWORKS',/,
     &5X,'THIS     PROGRAM     HAS   THE   ADDITIONAL      FACILITY     FOR:',/,
     &10X,'1     RANDOM    GENERATION      OF   A  NETWORK',/,
     &10X,'2     LINKING     WITH   DODIN'S     PROGRAM     TO  DETERMINE      DISTRI',/,
     &10X,'      BUTIONS     OF   NODE    REALIZATION       TIMES',/,
     &10X,'                                                        ',/,
     &10X,'NOTE        TO   RUN   AN COST.EXE,       YOU    MUST',//,
     &10X,'               1   CREATE    DATA    FILE   CONTAINING      GENERAL     INFO',/,
     &10X,'                   AND   DURATION/COST       INFORMATION      USING    THIS',/,
     &10X,'                   PROGRAM',/,
     &10X,'               2   RUN   DODTRNS.EXE        TO   DETERMINE      MEAN    AND   VAR',/,
     &10X,'                   OF   KEY   EVENTS     WHICH    ARE   PUT    IN   FILE    TRNS.AN',/,
     &10X,'                   (THIS    WILL    ALSO    GIVE    REAL TIME       LIST     OF   KEY',/,
     &10X,'                    EVENTS)',/,
     &10X,'               3   RUN   AN COST.EXE',///)
2      FORMAT(20X,'PROGRAM                              AN INPUT',//,
     &5X,'BY',
     &10X,'RUSSE                 S              VOGTMANN',
     &5X,'                                                    ',
     &5X,'ADVISOR        DR     S    E    E.MAGHRABY',///)
4      FORMAT(5X,'LIMITATIONS          OF   THE   PROGRAM',/
     &10X,'1     MAX   100   ARCS',/,
     &10X,'2     MAX   50   NODES',/,
     &10X,'3     MAX   30   DURATIONS     FOR   AN   ACTIVITY',/,
     &10X,'4     DIRECTED     ACYCLIC     NETWORKS     ONLY     ',/)
       RETURN
       END
C
C
```

```
C ***************************************************************
C
      SUBROUTINE     REED2(NANG)
C
C ***************************************************************
      CHARACTER     ISTOP*1
      WR. E(*,1)
 1    FORMAT(5X,'WOULD       YOU  LIKE   TO  GENERATE    A  RANDOM    NETWORK    ?',/,
     &5X,'IF    SO,   TYPE   IN    Y    ',/,
     &5X,'IF    NOT,   TYPE   IN   N   .THIS   MEANS   YOUR   NETWORK    IS  READY.',/,
     &5X,'TO    STOP,   TYPE   IN   S    ')
      READ(*,2)ISTOP
 2    FORMAT(A1)
      IF(ISTOP.EQ.'S'.OR.ISTOP.EQ.'s')              STOP
      IF(ISTOP.EQ.'Y'.OR.ISTOP.EQ.'y')              NANG=1
      IF(ISTOP.EQ.'N'.OR.ISTOP.EQ.'n')              NANG=0
      RETURN
      END
C
C
C ***************************************************************
C
      SUBROUTINE     REED3(NCONT)
C
C ***************************************************************
C
      CHARACTER     ISTOP*1
      WRITE(*,1)
 1    FORMAT(5X,'DO      YOU   HAVE   ACTIVITIES     WITH   CONTINUOUS     DISTRI ',/,
     &5X,'BUTIONS       ?',/,
     &5X,'IF    SO,   TYPE   IN    Y    ',/,
     &5X,'IF    NOT,   TYPE   IN    N    ',/,
     &5X,'TO    STOP,   TYPE   IN    S    ')
      READ(*,2)     ISTOP
 2    FORMAT(A1)
      IF(ISTOP.EQ.'S'.OR.ISTOP.EQ.'s')STOP
      IF(ISTOP.EQ.'Y'.OR.ISTOP.EQ.'y')              NCONT=[illegible]
      IF(ISTOP.EQ.'N'.OR.ISTOP.EQ.'n')              NCONT=[illegible]
      RETURN
      END
C
C ***************************************************************
C
      SUBROUTINE     REED5(N,M)
C
C ***************************************************************
C
      CHARACTER     ISTOP*1
 7    WRITE(*,1)
 1    FORMAT(5X,'TYPE         IN   THE   #  OF   NODES    AND    THE    #  OF   ARCS [illegible]
      READ(*,*)N,M
      IF(N.LE.50.AND.N.GE.2)          GO TO 3
```

```
      WRITE(*,2)
2     FORMAT(5X,'THE      # OF  NODES   IS  OUT   OF  RANGE.',/,
     &5X,'WOULD    YOU  LIKE   TO  TYPE   IN  THE  #  OF  NODES   AND   THE',/,
     &5X,'#   OF  ARCS   AGAIN   ?',/,
     &5X,'IF   SO,  TYPE   IN   Y   ',/,
     &5X,'IF   NOT,  TYPE   IN   S   AND   THE   PROGRAM   WILL   STOP.')
      READ(*,6)    ISTOP
6     FORMAT(A1)
      IF(ISTOP.EQ.'Y'.OR.ISTOP.EQ.'y')            THEN
         GO  TO  7
      ELSE
         STOP
      ENDIF
3     IF(M.GE.N-1.AND.M.LE.N*(N-1)/2.AND.M.LE.100)                GO  TO  8
      WRITE(*,4)
4     FORMAT(5X,'THE      NUMBER    OF  ARCS   IS  OUT   OF  RANGE.',/,
     &5X,'WOULD    YOU  LIKE   TO  TYPE   IN  THE  #  OF  NODES   AND   THE   ',/,
     &5X,'#   OF  ARCS   ?',/,
     &5X,'IF   SO,  TYPE   IN   Y   ',/,
     &5X,'IF   NOT,  TYPE   IN   S   AND   THE   PROGRAM   WILL   STOP.')
      READ(*,6)    ISTOP
      IF(ISTOP.EQ.'Y'.OR.ISTOP.EQ.'y')            THEN
         GO  TO  7
      ELSE
         STOP
      ENDIF
8     RETURN
      END
C
C
C   ***************************************************************
C
      SUBROUTINE    REED6(SCAL)
C
C   ***************************************************************
C
      CHARACTER    ISTOP*1
[illegible] WRITE(*,[illegible])
[illegible] FORMAT(5X,'TYPE     IN  THE   INTERVAL    WIDTH   TO  BE  USED   IN [illegible]
     &5X,'DISCRETIZATION       FOR  OBTAINING    MEAN   AND   VAR [illegible]
     &5X,'A   TYPICAL    VALUE   IS [illegible]
      READ(*,*)    SCAL
      IF(SCAL.[illegible])         THEN
         WRITE(*,2)
2        FORMAT(5X,'SCAL      VALUE    INCORRECT [illegible]
     &5X,'WOULD    YOU  LIKE   TO  INPUT   CORRECT    VALUE [illegible]
     &5X,'IF   SO,  TYPE   IN  Y [illegible]
     &5X,'IF   NOT,  TYPE   IN  N  AND   THE  PROGRAM   W[illegible]
         READ(*,[illegible])   [illegible]
[illegible]      FORMAT(A[illegible])
         [illegible]                    THEN
            [illegible]
         ELSE
```

```
            STOP
         ENDIF
      ELSE
      ENDIF
      RETURN
      END
C
C
C ********************************************************************
C
      SUBROUTINE     REED7(KEY,KEYN,NRR,N)
C
C ********************************************************************
C
      DIMENSION     KEYN(50)
      CHARACTER     ISTOP*1,OK*1
 4    WRITE(*,1)
 1    FORMAT(5X,'TYPE      IN  THE   NUMBER   OF  ORDERED   PAIRS   IN  THE   ',/,
     &5X,'APPROXIMATE      DISTRIBUTION.',/)
      READ(*,*)NRR
      IF(NRR.GT.30.OR.NRR.LE.0)          THEN
         WRITE(*,8)
 8    FORMAT(5X,'VALUE      OF  NRR   OUT   OF  RANGE',/,
     &5X,'WOULD     YOU  LIKE   TO  INPUT   CORRECTED    VALUE   ?',/,
     &5X,'IF    SO  TYPE   IN  Y  ',/,
     &5X,'IF    NOT,  TYPE   IN  N  AND   THE   PROGRAM    WILL   STOP.')
         READ(*,7)ISTOP
 7    FORMAT(A1)
         IF(ISTOP.EQ.'Y'.OR.ISTOP.EQ.'y')             THEN
            GO  TO  4
         ELSE
            STOP
         ENDIF
      ELSE
      ENDIF
C
 5    WRITE(*,2)
 2    FORMAT(5X,'TYPE      IN  THE  #  OF  MILESTONES    OR  KEY   NODES   IN',/,
     &5X,'THE     PROJECT.')
      READ(*,*)KEY
      IF(KEY.LT.1.OR.KEY.GT.N.OR.KEY.GT.50)              THEN
         WRITE(*,9)
 9    FORMAT(5X,'VALUE      OF  KEY   OUT   OF  RANGE',/,
     &5X,'WOULD     YOU   LIKE   TO  TYPE   IN   CORRECTED    VALUE   ?',/,
     &5X,'IF    SO,  TYPE   IN  Y  ',/,
     &5X,'IF    NOT,  TYPE   IN  N  AND   THE   PROGRAM    WILL   STOP.')
         READ(*,7)     ISTOP
         IF(ISTOP.EQ.'Y'.OR.ISTOP.EQ.'y')              THEN
            GO  TO  5
         ELSE
            STOP
         ENDIF
      ELSE
```

```
      ENDIF
C
 6    WRITE(*,3)
 3    FORMAT(5X,'TYPE    IN  THE  LIST  OF  KEY  NODES  IN  ASCENDING   ORDER.',/)
      READ(*,*)(KEYN(I),I=1,KEY)
      OK='Y'
      IF(KEYN(1).LE.0)     OK='N'
      IF(KEYN(KEY).NE.N)     OK='N'
      DO  11  I=2,KEY
      J=I-1
       IF(KEYN(J).GE.KEYN(I))      OK='N'
 11   CONTINUE
      IF(OK.EQ.'N'.OR.OK.EQ.'n')       THEN
         WRITE(*,10)
 10   FORMAT(5X,'VALUES    OF  KEY  NODES  INCORRECT',/,
     &5X,'EITHER   FIRST  KEY  NODE  IS  LE  0',/,
     &5X,'OR   LAST  KEY  NODE  IS  NE  N',/,
     &5X,'OR   KEY  NODES  NOT  IN  ASCENDING   ORDER',/,
     &5X,'WOULD   YOU  LIKE  TO  INPUT  THE  NODES  AGAIN  ?',/,
     &5X,'IF   SO,  TYPE  IN  Y  ',/,
     &5X,'IF   NOT  TYPE  IN  N  AND  THE  PROGRAM   WILL  STOP.')
         READ(*,7)   ISTOP
         IF(ISTOP.EQ.'Y'.OR.ISTOP.EQ.'y')       THEN
            GO  TO  6
         ELSE
            STOP
         ENDIF
      ELSE
      ENDIF
      RETURN
      END
C
C
C ***************************************************************
C
      SUBROUTINE    HALT
C
C ***************************************************************
C
      CHARACTER    ISTOP*1
      WRITE(*,1)
 1    FORMAT(5X,'WOLUD     YOU  LIKE  TO  CONTINUE   OR  WOULD  YOU  LIKE',/,
     &5X,'TO   STOP  ?',/,
     &5X,'TO   CONTINUE,TYPE    IN   Y   ',/,
     &5X,'TO   STOP,  TYPE  IN   S   ')
      READ(*,2)    ISTOP
 2    FORMAT(A1)
      IF(ISTOP.EQ.'S'.OR.ISTOP.EQ.'s')       STOP
      RETURN
      END
C
C
C ***************************************************************
```

```
C
      SUBROUTINE   REED8(NOI)
C
C **************************************************************
C
      CHARACTER   ISTOP*1
 3    WRITE(*,1)
 1    FORMAT(5X,'IT   IS ASSUMED  THAT  THE  ACTIVITY  OF THE  NETWORK',/,
     &5X,'CAN  BE DIVIDED  INTO  SEVERAL  PARTITIONS.  EACH  PARTITION',/,
     &5X,'IS  REPRESENTED  BY AN ACTIVITY  NUMBER  WHICH  IS THE  ',/,
     &5X,'UPPER  LIMIT  OF THE  PARTITION.  EACH  PARTITION  HAS A',/,
     &5X,'DIFFERENT  TYPE  OF DISTRIBUTION.',//,
     &5X,'TYPE  IN THE  NUMBER  OF PARTITIONS')
      READ(*,*)   NOI
      IF(NOI.LE.0)   THEN
      WRITE(*,2)
 2    FORMAT(5X,'THE   NUMBER  OF PARTITIONS  IS NOT  A POSITIVE   ',/,
     &5X,'NUMBER.',/,
     &5X,'WOULD  YOU  LIKE  TO RE-ENTER  THE  NUMBER  OF PARTITIONS',/,
     &5X,'IF  SO,  TYPE  IN  Y  ',/,
     &5X,'IF  NOT,  TYPE  IN  S  AND  THE  PROGRAM  WILL  STOP.')
      READ(*,4)ISTOP
 4    FORMAT(A1)
          IF(ISTOP.EQ.'Y'.OR.ISTOP.EQ.'y')        THEN
               GO TO 3
          ELSE
               STOP
          ENDIF
      ELSE
      ENDIF
 5    RETURN
      END
C
C
C **************************************************************
C
      SUBROUTINE   REED9(I,NULT,NDS,NT)
C
C **************************************************************
C
      CHARACTER   ISTOP*1,   OK*1
 4    WRITE(*,1)
 1    FORMAT(5X,'NEXT   FOR  EACH  PARTITION,  TYPE  IN  THE  FOLLOWING',/,
     &5X,'3  QUANTITIES  IN SERIAL  ORDER:',/,
     &5X,'   A.  NULT:  IT  IS THE  NUMBER  OF THE  ACTIVITY  REPRESENT-',/,
     &5X,'             ING  THE  UPPER  LIMIT  OF THE  PARTITION.',/,
     &5X,'   B.  NDS  : IT  IS THE  TYPE  OF DISTRIBUTION  OF THE  ',/,
     &5X,'            ACTIVITY  DURATIONS  IN THIS  PARTITION.',/,
     &5X,'              FOR  UNIFORM  DISTN.,      NDS=1',/,
     &5X,'              FOR  TRIANGULAR  DISTN.,   NDS=2',/,
     &5X,'              FOR  NORMAL  DISTN.,       NDS=3',/,
     &5X,'              FOR  EXPONENTIAL  DISTN.,NDS=4',/,
     &5X,'              FOR  GAMMA  DISTN.,        NDS=5',/,
```

```
     &5X,'                    FOR   BETA   DISTN.,          NDS=6',/,
     &5X,'                    FOR   DISCRETE    DISTN.,      NDS=7',/,
     &5X,'                    NDS=7   FOR   ANY   DISTN.   REPRESENTED     BY',/,
     &5X,'                    A  SET   OF   FINITE    ORDERED    PAIRS',//,
     &5X,'    C.  NT    :  IT   IS  THE   #  OF   ORDERED    PAIRS    IF  NDS=7',/,
     &5X,'                  IF   NDS   IS   NOT   EQUAL    TO  7,  NT=0')
      WRITE(*,3)     I
 3    FORMAT(5X,'TYPE      IN   THE   VALUES    OF   NULT,   NDS   AND   NT',/,
     &5X,'FOR     THE   PARTITION     NO.',I2)
      READ(*,*)NULT,NDS,NT
      CALL   CHECK4(NULT,OK,ISTOP)
      IF(OK.EQ.'N'.OR.OK.EQ.'n')           THEN
          IF(ISTOP.EQ.'Y'.OR.ISTOP.EQ.'y')             THEN
              GO  TO  4
          ELSE
              STOP
          ENDIF
      ELSE
      ENDIF
      CALL   CHECK3(NDS,NT,OK,STOP)
      IF(OK.EQ.'N'.OR.OK.EQ.'n')           THEN
         IF(ISTOP.EQ.'Y'.OR.ISTOP.EQ.'y')             THEN
            GO  TO  4
         ELSE
            STOP
         ENDIF
      ELSE
      ENDIF
      RETURN
      END
C
C
C ****************************************************************
C
      SUBROUTINE     REED10(I,IT,EX,STDX,VMIN,VMAX)
C
C ****************************************************************
C
      CHARACTER     ISTOP*1,     OK*1
 4    WRITE(*,1)     I
 1    FORMAT(5X,'TYPE      IN   THE   VALUES    OF   EX,   STDX,',/,
     &5X,'VMIN,      VMAX,    FOR   THE   ACTIVITIES     IN  PARTITION     NO.   ',I2,/,
     &10X,'THE      PARAMETERS      ARE   EXPLAINED     BELOW:',//,
     &5X,'DISTRIBUTION        INDEX     EX    STDX    REMARKS',//,
     &5X,'     UNIFORM             1     0.0    0.0              ',/,
     &5X,'TRIANGULAR               2       M    0.0              ',/,
     &5X,'     NORMAL              3    MEAN    S.D.             ',/,
     &5X,'   EXPONENTIAL           4    MEAN                     ',/,
     &5X,'       GAMMMA            5    ALPHA   BETA             ',/,
     &5X,'        BETA             6    ALPHA   BETA             ',/,
     &5X,'VMIN      IS   THE   MINIMUM    VALUE    OF   THE   RANDOM    VARIABLE.',/,
     &5X,'IT     IS  USUALLY    DECIDED     BY   THE   CRITERION:-',/,
     &5X,'P(Y.LE.VMIN)=0.0005.',/,
```

```
     &5X,'VMAX    IS  THE  MAXIMUM   VALUE   OF  THE  RANDOM   VARIABLE.',/,
     &5X,'IT   IS  DECIDED   USUALLY   BY  THE  CRITERION:-',/,
     &5X,'P(Y.GE.VMAX)=0.0005.')
      READ(*,*)     EX,STDX,VMIN,VMAX
      CALL   CHECK1(IT,EX,STDX,VMIN,VMAX,OK,ISTOP)
      IF(OK.EQ.'N'.OR.OK.EQ.'n')          THEN
          IF(ISTOP.EQ.'Y'.OR.ISTOP.EQ.'y')            THEN
              GO  TO  4
          ELSE
              STOP
          ENDIF
      ELSE
      ENDIF
      RETURN
      END
C
C
C *****************************************************************
C
      SUBROUTINE     REED11(I,J,R,PR)
C
C *****************************************************************
C
      CHARACTER    ISTOP*1
 3    WRITE(*,1)     J,I
 1    FORMAT(5X,'TYPE       IN  THE',I2,'TH     PAIR   OF  ACTIVITY    DURATION',/,
     &5X,'AND    PROBABILITY     FOR  PARTITION    NO.   ',I2,/)
      READ(*,*)     R,PR
      IF(R.LE.0.0.OR.PR.LT.0.0.OR.PR.GT.1.0)             THEN
         WRITE(*,2)
   2     FORMAT(5X,'THE      VALUES   OF  DURATION    AND/OR    PROBABILITY',/,
     &5X,'ARE    OUT   OF  RANGE',/,
     &5X,'WOULD     YOU  LIKE   TO  INPUT   CORRECTED    VALUES    ?  ',/,
     &5X,'IF    SO,   TYPE   IN  Y',/,
     &5X,'IF    NOT,   TYPE   IN  S  AND   THE   PROGRAM    WILL   STOP.')
         READ(*,4)     ISTOP
 4       FORMAT(A1)
         IF(ISTOP.EQ.'Y'.OR.ISTOP.EQ.'y')           THEN
             GO  TO  3
         ELSE
             STOP
         ENDIF
      ELSE
      ENDIF
      RETURN
      END
C
C
C *****************************************************************
C
      SUBROUTINE      [illegible](I,NS,NE,N,S[illegible],NR)
C
C *****************************************************************
```

```
C
      CHARACTER    ISTOP*1,    OK*1
 6    WRITE(*,1)     I
 1    FORMAT(5X,'FOR      ACTIVITY    ',I2,'    READ   IN   THE   PARAMETERS',/,
     &5X,'NS,    NE,   NDSTT,    NR   IN   THAT   ORDER:',//,
     &5X,'    A.  NS    :  STARTING    NODE',/,
     &5X,'B.       NE    :  ENDING    NODE',/,
     &5X,'C.     NDSTT:    IT   IS  THE   TYPE   OF  DISTRIBUTION     OF  THE   ',/,
     &5X,'                DURATION     OF   THE   ACTIVITY.',/,
     &5X,'                   FOR   UNIFORM    DISTN.,         NDSTT=1',/,
     &5X,'                   FOR   TRIANGULAR     DISTN.,     NDSTT=2',/,
     &5X,'                   FOR   NORMAL    DISTN.,          NDSTT=3',/,
     &5X,'                   FOR   EXPONENTIAL     DISTN.,NDSTT=4',/,
     &5X,'                   FOR   GAMMA    DISTN.,           NDSTT=5',/,
     &5X,'                   FOR   BETA   DISTN.,             NDSTT=6',/,
     &5X,                    FOR   DISCRETE    DISTN.,        NDSTT=7',/,
     &5X,'                   NDS=7    FOR   ANY   DISTN.    REPRESENTED     BY',/,
     &5X,'                   A  SET   OF  FINITE    ORDERED    PAIRS,'//,
     &5X,'    C.  NR    :  IT   IS  THE   #  OF  ORDERED    PAIRS    IF   NDSTT=7',/,
     &5X,'              IF   NDSTT    IS   NOT   EQUAL    TO  7,   NR=0')
      READ(*,*)     NS,NE,NDSTT,NR
      CALL    CHECK2(NS,NE,OK,ISTOP)
      IF(OK.EQ.'N'.OR.OK.EQ.'n')           THEN
          IF(ISTOP.EQ.'Y'.OR.ISTOP.EQ.'y')              THEN
              GO  TO  6
          ELSE
              STOP
          ENDIF
      ELSE
      ENDIF
       CALL    CHECK3(NDSTT,NR,OK,ISTOP)
      [illegible]                                        THEN
          [illegible]                                                 THEN
              [illegible]
          ELSE
              [illegible]
          [illegible]
      [illegible]
      [illegible]
       [illegible]
      [illegible]
```

....

[illegible]

....

[illegible]

```
&10X,'THE     PARAMETERS     ARE   EXPLAINED     BELOW:',//,
&5X,'DISTRIBUTION        INDEX     EX    STDX     REMARKS',//,
&5X,'     UNIFORM           1     0.0    0.0                 ',/,
&5X,'TRIANGULAR             2       M    0.0                 ',/,
&5X,'     NORMAL            3    MEAN   S.D.                ',/,
&5X,'   EXPONENTIAL         4    MEAN                        ',/,
&5X,'       GAMMMA          5    ALPHA   BETA                ',/,
&5X,'        BETA           6    ALPHA   BETA                ',/,
&5X,'VMIN     IS  THE   MINIMUM    VALUE   OF   THE   RANDOM    VARIABLE.',/,
&5X,'IT     IS  USUALLY    DECIDED    BY  THE   CRITERION:-',/,
&5X,'P(Y.LE.VMIN)=0.0005.',/,
&5X,'VMAX     IS  THE   MAXIMUM    VALUE   OF   THE   RANDOM    VARIABLE.',/,
&5X,'IT     IS  DECIDED    USUALLY    BY  THE   CRITERION:-',/,
&5X,'P(Y.GE.VMAX)=0.0005.')
 READ(*,*)     EX,STDX,VMIN,VMAX
 CALL   CHECK1(IT,EX,STDX,VMIN,VMAX,OK,ISTOP)
 IF(OK.EQ.'N'.OR.OK.EQ.'n')            THEN
      IF(ISTOP.EQ.'Y'.OR.ISTOP.EQ.'y')             THEN
           GO  TO  4
      ELSE
           STOP
      ENDIF
 ELSE
 ENDIF
 RETURN
 END
```

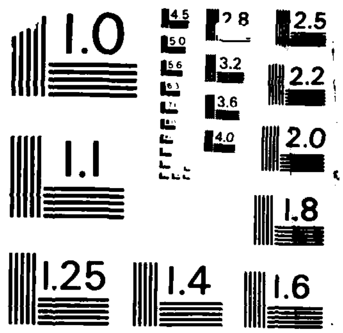
```
****************************************************************

  SUBROUTINE   [illegible]

****************************************************************

  CHARACTER    [illegible]
[illegible]
```

```
      ELSE
      ENDIF
      RETURN
      END
C
C ************************************************************
C
      SUBROUTINE    REED16(I,NS,NE,NDSTT,FC,DM)
C
C ************************************************************
C
 6    WRITE(*,1)    I,NS,NE
 1    FORMAT(5X,'FOR     ACTIVITY    ',I2,'    FROM   NODE   ',I3,'TO    NODE   ',I3,/,
     &5X,'INPUT     THE   FOLLOWING     PARAMETERS:',//,
     &5X,'A.       FC    :  FIXED    COST    FOR   ACTIVITY',/,
     &5X,'B.       DM    :  CONSTANT     MULTIPLIER     ON   DURATION',/,
     &5X,'C.     NDSTT:    IT   IS   THE   TYPE   OF   DISTRIBUTION     OF   THE   ',/,
     &5X,'                 COST    ABOUT    <Y*DM>.',/,
     &5X,'                   FOR   CONSTANT*DUR.,             NDSTT=1',/,
     &5X,'                   FOR   NORMAL    DISTN.,          NDSTT=2',/,
     &5X,'                   FOR   UNIFORM    DISTN.,         NDSTT=3',/,
     &5X,'                   FOR   TRIANGULAR     DIST.,      NDSTT=4',/)
      READ(*,*)     FC,DM,NDSTT
      IF  (FC.LT.0.0.OR.DM.LT.0.0)          GOTO   6
      IF  (NDSTT.LT.1.OR.NDSTT.GT.4)          GOTO   6
C
      RETURN
      END
C
C
C ************************************************************
C
      SUBROUTINE    REED17(I,P1,VMIN,VMAX)
C
C ************************************************************
C
 4    WRITE(*,1)    I
 1    FORMAT(5X,'TYPE       IN   THE   VALUES    OF   P1',/,
     &5X,'VMIN,      VMAX,    FOR   THE   ACTIVITIY     NO.   ',I2,/,
     &10X,'THE     PARAMETERS     ARE   EXPLAINED     BELOW:',//,
     &5X,'DISTRIBUTION        INDEX     P1               ',//,
     &5X,'    CONSTANT*Y          1      0.0              ',/,
     &5X,'      NORMAL            2      VAR              ',/,
     &5X,'      UNIFORM           3      0.0              ',/,
     &5X,'      TRIANGULAR        4      0.0              ',/,
     &5X,'VMIN      IS   THE   MINIMUM    VALUE   OF   THE   RANDOM    VARIABLE.',/,
     &5X,'  [illegible]   IS   USUALLY    DECIDED    BY   THE   CRITERION: [illegible]
     &5X, [illegible] VMIN [illegible] .0005 [illegible]
     &5X,'VMAX      IS   THE   MAXIMUM    VALUE   OF   THE   RANDOM    VARIABLE.[illegible]
     &5X,  [illegible]   IS   DECIDED    USUALLY    BY   THE   CRITERION [illegible]
     &5X, [illegible] VMAX [illegible] .0005 [illegible]
      READ(*,*)     P1,VMIN,VMAX
      [illegible]
```

```
      RETURN
      END
C
C ************************************************************
C
      SUBROUTINE    REED18(I,FC,DM,NDS)
C
C ************************************************************
C
      CHARACTER    ISTOP*1,    OK*1
 4    WRITE(*,1)
 1    FORMAT(5X,'NEXT      FOR  EACH  PARTITION,    TYPE  IN  THE  FOLLOWING',/,
     &5X,'3  QUANTITIES    IN  SERIAL   ORDER:',/,
     &5X,'   A.  FC    :  FIXED   COST   FOR  ACTIVITIES    IN  THIS  PAR-',/,
     &5X,'                TITION.',/,
     &5X,'   B.  DM    :  CONSTANT    MULTIPLIER    ON  DURATION',/,
     &5X,'                (DURATION*DM     IS  THE  MEAN   OF  THE   COST',/,
     &5X,'                 DISTRIBUTION.',/,
     &5X,'   C.  NDS   :  IT  IS  THE   TYPE  OF  DISTRIBUTION     OF  THE',/,
     &5X,'                COST   ABOUT   <DM>',/,
     &5X,'                 FOR  CONSTANT*DUR.,          NDS=1',/,
     &5X,'                 FOR  NORMAL    DIST.,        NDS=2',/,
     &5X,'                 FOR  UNIFORM    DISTN.,      NDS=3',/,
     &5X,'                 FOR  TRIANGULAR     DIST.,   NDS=4',////)
      WRITE(*,3)    I
 3    FORMAT(5X,'TYPE     IN  THE   VALUES   OF  FC,  DM,  AND  NDS',/,
     &5X,'FOR    THE   PARTITION    NO.',I2)
      READ(*,*)FC,DM,NDS
      IF(NDS.GT.4.OR.NDS.LT.1)         GOTO  4
      IF(FC.LT.0.0.OR.DM.LT.0.0)         GOTO  4
      RETURN
      END
C
C
C ************************************************************
C
      SUBROUTINE    REED19(I,P1,VMIN,VMAX)
C
C ************************************************************
C
 4    WRITE(*,1)    I
 1    FORMAT(5X,'TYPE       IN  THE   VALUES [illegible]
     &5X,'VMIN,     VMAX,    FOR   THE   A[illegible]
     &10X,'THE     PARAMETERS     ARE   [illegible]
     &5X,'DISTRIBUTION        [illegible]
     &5X,'     CONST[illegible]
     &5X,'     NORMAL [illegible]
     &5X,'     UNIFORM [illegible]
     &5X,'     [illegible]
     &5X,'VMIN [illegible]
     &5X,[illegible]
     &5X,[illegible]
     &5X,'VMAX [illegible]
```

AD-A176 909 PROJECT BIDDING UNDER CHANCE TIME ESTIMATES(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH R S VOGTMANN 1986 AFIT/CI/NR-87-32T 2/2

UNCLASSIFIED F/G 5/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
     &5X,'IT    IS  DECIDED    USUALLY    BY  THE   CRITERION:-',/,
     &5X,'P(Y.GE.VMAX)=0.0005.')
      READ(*,*)     P1,VMIN,VMAX
      IF  (P1.LT.0.0.OR.VMIN.LT.0.0.OR.VMAX.LT.VMIN)                GOTO   4
      RETURN
      END
C
C
C *****************************************************************
C
      SUBROUTINE     CHECK1(IT,EX,STDX,VMIN,VMAX,OK,ISTOP)
C
C ****************************************************************
C
      CHARACTER     ISTOP*1,OK*1
      OK='Y'
      IF(VMIN.LE.0.0.OR.VMAX.LE.VMIN)            THEN
         OK='N'
      ELSE   IF(IT.EQ.3)      THEN
         IF(EX.LE.VMIN.OR.EX.GE.VMAX.OR.STDX.LE.0.0)               THEN
            OK='N'
         ELSE
         ENDIF
      ELSE   IF(IT.EQ.4)      THEN
         IF(EX.LE.VMIN.OR.EX.GE.VMAX)            THEN
            OK='N'
         ELSE
         ENDIF
      ELSE   IF(IT.EQ.5.OR.IT.EQ.6)         THEN
         IF(EX.LE.0.0.OR.STDX.LE.0.0)            THEN
            OK='N'
         ELSE
         ENDIF
      ELSE
      ENDIF
C
      IF(OK.EQ.'N'.OR.OK.EQ.'n')           THEN
         WRITE(*,1)
 1    FORMAT(5X,'SOME       OF  THE   VALUES   ARE  OUT  OF  RANGE',/,
     &5X,'WOULD     YOU  LIKE   TO  INPUT   THE  CORRECTED    VALUES   ?',/,
     &5X,'IF    SO,  TYPE   IN  Y  ',/,
     &5X,'IF    NOT,   TYPE   IN  N  AND   THE  PROGRAM   WILL   STOP.')
         READ(*,2)     ISTOP
 2    FORMAT(A1)
      ELSE
      ENDIF
      RETURN
      END
C
C
C *******************************************************************
C
      SUBROUTINE     CHECK2(NS,NE,OK,ISTOP)
```

```
C
C ******************************************************************
C
      CHARACTER    OK*1,   ISTOP*1
      OK='Y'
      IF(NS.LE.0.OR.NE.LE.0.OR.NE.LE.NS)            THEN
         OK='N'
      ELSE
      ENDIF
C
      IF(OK.EQ.'N'.OR.OK.EQ.'n')          THEN
         WRITE(*,1)
 1    FORMAT(5X,'STARTING        AND/OR    ENDING    NODES    IS  INCORRECT',/,
     &5X,'WOULD      YOU   LIKE    TO  INPUT    THE   CORRECT    VALUES    ?',/,
     &5X,'IF    SO,   TYPE    IN  Y  ',/,
     &5X,'IF    NOT,   TYPE    IN  N  AND   THE   PROGRAM    WILL   STOP.')
        READ(*,2)     ISTOP
 2    FORMAT(A1)
      ELSE
      ENDIF
      RETURN
      END
C
C
C ******************************************************************
C
      SUBROUTINE     CHECK3(NDS,NR,OK,ISTOP)
C
C ******************************************************************
C
      CHARACTER    OK*1,   ISTOP*1
      OK='Y'
      IF(NDS.LT.1.OR.NDS.GT.7)          THEN
         OK='N'
      ELSE   IF   (NDS.EQ.7.AND.NR.EQ.0)          THEN
         OK='N'
      ELSE
      ENDIF
C
      IF(OK.EQ.'N'.OR.OK.EQ.'n')          THEN
         WRITE(*,1)
 1    FORMAT(5X,'VALUES        OF  DISTRIBUTION      TYPE   AND/OR     ',/,
     &5X,'NO.    OF   PAIRS    IS   INCORRECT',/,
     &5X,'WOULD      YOU   LIKE    TO  INPUT    THE   CORRECT    VALUES    ?',/,
     &5X,'IF    SO,   TYPE    IN  Y  ',/,
     &5X,'IF    NOT,   TYPE    IN  N  AND   THE   PROGRAM    WILL   STOP.')
        READ(*,2)     ISTOP
 2    FORMAT(A1)
      ELSE
      ENDIF
      RETURN
      END
C
```

```
C
C ********************************************************************
C
      SUBROUTINE     CHECK4(NULT,OK,ISTOP)
C
C ********************************************************************
C
      CHARACTER     OK*1,    ISTOP*1
      OK='Y'
      IF(NULT.LE.0)       THEN
         OK='N'
      ELSE
      ENDIF
C
      IF(OK.EQ.'N'.OR.OK.EQ.'n')            THEN
         WRITE(*,1)
 1    FORMAT(5X,'VALUE        OF   NULT   IS   INCORRECT     ',/,
     &5X,'WOULD      YOU   LIKE   TO  INPUT   THE   CORRECT    VALUES    ?',/,
     &5X,'IF    SO,   TYPE   IN  Y  ',/,
     &5X,'IF    NOT,   TYPE   IN   N  AND   THE   PROGRAM    WILL   STOP.')
        READ(*,2)     ISTOP
 2    FORMAT(A1)
      ELSE
      ENDIF
      RETURN
      END
C **********************************************************************
C
      SUBROUTINE     GNRAT
C     RANDOM    ACTIVITY    NETWORK    GENERATOR.     THE   AN   HAS   N  NODES   AND   M  ARCS.
C     IT   IS   OF   TYPE   ACTIVITY    ON  ARC.

C **********************************************************************
      REAL*8    IX
      COMMON    /PAR1/N,M,NRR,NCONT,MCS,NSIM,SCAL,NR(100),NT(50)
      COMMON    /PAR2/NS(100),NE(100),NDSTT(100),R(100,30),PR(100,30)
      COMMON    /PARS/IND(200),NCF(200),NCS(200),IIN(60),IOUT(60)
      DIMENSION     NAF(60),NBE(60),A(50,50)
      IX=12345.0D0
      WRITE(*,4)N,M
 4    FORMAT('1',6X,'GENERATE          AN   ACTIVITY    NETWORK    WITH',I3,'     NODES    AND',
     *I4,'    ARCS')
      DO  53   I=1,N
      DO  52   J=1,N
 52   A(I,J)=0.0
 53   CONTINUE
      L=N*(N-1)/2
      N1=N-1
      N2=N-2
      DN=FLOAT(N)
      LM=L-M
      MF=2*N-4
      DL=FLOAT(L)/2.0+1.0
```

```
      DN2=DN*(DN-1.0)
      DN3=DN+0.5
      IF(DL-M)10,10,20
C     THE DELETION METHOD
 10   DO 3 I=1,N1
      JJ=I+1
      DO 2 J=JJ,N
 2    A(I,J)=1.0
      IIN(I)=I-1
 3    IOUT(I)=N-I
      IIN(N)=N-1
 48   DO 13 I=1,LM
 11   CALL RAND(IX,Y)
      Y1=Y*DN2+0.25
      X=DN3-SQRT(Y1)
      NO=IFIX(X)
      IF(NO.GT.X)   NO=NO-1
      IF(IOUT(NO).LT.2)   GO TO 11
      K=0
      JJ=NO+1
      DO 22 J=JJ,N
      IF(IIN(J).LT.2)   GO TO 22
      IF(A(NO,J).EQ.0.0)   GO TO 22
      K=K+1
      NAF(K)=J
 22   CONTINUE
      IF(K.EQ.0)   GO TO 11
      DEN=1.0/K
      CALL RAND(IX,X)
      DO 23 J=1,K
      UP=DEN*J
      IF(X.GT.UP)   GO TO 23
      NI=NAF(J)
      GO TO 5
 23   CONTINUE
 5    A(NO,NI)=0.0
      IIN(NI)=IIN(NI)-1
      IOUT(NO)=IOUT(NO)-1
 13   CONTINUE
      GO TO 50
C     THE ADDITION METHOD
 20   DO 30 I=1,N1
      IOUT(I)=0
 30   IIN(I)=0
      NRC=N-3
      NEM=N-3
      A(1,2)=1.0
      A(N1,N)=1.0
      NARCS=2
      IOUT(1)=1
      IOUT(N1)=1
      IIN(2)=1
      IIN(N)=1
```

```
      IF(MF.GE.M)      GO  TO  40
      KK=0
31    CALL   RAND(IX,Y)
      Y1=Y*DN2+0.25
      X=DN3-SQRT(Y1)
      NO=IFIX(X)
      IF(NO.GT.X)      NO=NO-1
      NN=N-NO
      IF(IOUT(NO).GE.NN)       GO  TO  31
      K=0
      JJ=NO+1
      DO  32  J=JJ,N
      IF(A(NO,J).EQ.1.0)       GO  TO  32
      K=K+1
      NAF(K)=J
32    CONTINUE
      DEN=1.0/K
      CALL   RAND(IX,X)
      UP=0.0
      DO  33  J=1,K
      UP=UP+DEN
      IF(X.GT.UP)      GO  TO  33
      NI=NAF(J)
      GO  TO  34
33    CONTINUE
34    A(NO,NI)=1.0
      NARCS=NARCS+1
      IF(IOUT(NO).EQ.0)        NEM=NEM-1
      IF(IIN(NI).EQ.0)       NRC=NRC-1
      IIN(NI)=IIN(NI)+1
      IOUT(NO)=IOUT(NO)+1
      IF(NARCS.GE.M)       GO  TO  50
      IF(KK.EQ.1)      GO  TO  31
      MF=M-NARCS-NRC-NEM
      IF(MF.GT.0)      GO  TO  31
40    IF(NRC.EQ.0)      GO  TO  45
      K=0
      DO  41  I=3,N1
      IF(IIN(I).GT.0)       GO  TO  41
      K=K+1
      NAF(K)=I
41    CONTINUE
      IF(K.EQ.0)      GO  TO  45
      DO  42  I=1,K
      IJ=K+1-I
      NI=NAF(IJ)
      CALL   RAND(IX,Y)
      X=1.0+(NI-1)*Y
      NO=IFIX(X)
      IF(NO.GT.X)      NO=NO-1
      A(NO,NI)=1.0
      NARCS=NARCS+1
      IIN(NI)=IIN(NI)+1
```

```
 42     IOUT(NO)=IOUT(NO)+1
 45     IF(NEM.EQ.0)     GO  TO  51
        K=0
        DO  43  I=2,N2
        IF(IOUT(I).GT.0)      GO  TO  43
        K=K+1
        NBE(K)=I
 43     CONTINUE
        IF(K.EQ.0)     GO  TO  51
        DO  44  I=1,K
        NO=NBE(I)
        CALL   RAND(IX,X)
        Y=NO+1+(N-NO)*X
        NI=IFIX(Y)
        IF(NI.GT.Y)     NI=NI-1
        IIN(NI)=IIN(NI)+1
        IOUT(NO)=IOUT(NO)+1
        NARCS=NARCS+1
 44     A(NO,NI)=1.0
 51     KK=1
        IF(NARCS-M)31,50,47
 47     LM=NARCS-M
        GO  TO  48
 50     K=0
        WRITE(*,14)NARCS
 14     FORMAT(6X,'THE      ACTIVITY    NETWORK    HAS   THE   FOLOWING    ',I3,'    ARCS')
        WRITE(*,15)
 15     FORMAT(/,5X,'             I    NS(I)     NE(I)')
        DO  17  I=1,N1
        JJ=I+1
        DO  16  J=JJ,N
        IF(A(I,J).NE.1.0)      GO  TO  16
        K=K+1
        NS(K)=I
        NE(K)=J
        WRITE(*,8)K,NS(K),NE(K)
 8      FORMAT(5X,5I7)
 16     CONTINUE
 17     CONTINUE
        RETURN
        END
C
C  ********************************************************************
C
        SUBROUTINE     RAND(IX,D)
C
C  ********************************************************************
C
        DOUBLE    PRECISION     Y,A,P,IX,B15,B16,XHI,XALO,LEFTLO,FHI,K
C
        DATA   A/16807.D0/,B15/32768.D0/,B16/65536.D0/,P/2147483647.D0/
C
        XHI=IX/B16
```

```
      XHI=XHI-DMOD(XHI,1.D0)
      XALO=(IX-XHI*B16)*A
      LEFTLO=XALO/B16
      LEFTLO=LEFTLO-DMOD(LEFTLO,1.D0)
      FHI=XHI*A+LEFTLO
      K=FHI/B15
      K=K-DMOD(K,1.D0)
      IX=(((XALO-LEFTLO*B16)-P)+(FHI-K*B15)*B16)+K
      IF(IX.LT.0.D0)IX=IX+P
      Y=IX*4.656612875E-10
      D=REAL(Y)
      IX=IX+1.0
      RETURN
      END
C     ***********************************************************************
```

Appendix 9.2. Logic of Program AN-COST

1. OBTAIN NECESSARY NETWORK INFORMATION

1.1 CALL subroutine INTRO containing introduction and user information.

1.2 Obtain input and output file names from user.

1.3 Initialize variables to zero.

1.4 Read in network/duration/cost data from input file.

1.5 Obtain variable values necessary for MCS from user (% project cost to receive at project start, inflation rate, random number seed, and period length).

1.6 Read in mean and std.dev. values of key events from file TRNS.ANC

2. DETERMINE COST ALLOCATION PROPORTIONS

2.1 CALL subroutine SUBNET to determine the subgraphs of key events.

- find subnetworks of all key events
- remove activities from subnetworks of key event i that are members of the subnetwork of key event j, where j : i

2.2 CALL subroutine SEPARATE to determine allocation Proportions, Prop(i), of common act.'s

- DO over activities
  - IF common to more then one key event THEN
    - determine cost allocation proportions via probability method or user determined percentages.
    - IF user determined THEN
      - obtain percentages from user
    - ELSE IF probability method THEN
      - determine key event with smallest mean
      - find p(i) = Pr { X(smallest) \ X(i) }
        - IF p(i) ^ 0.9 " set Prop(i)=0.00
      - for remaining key events, set w(i)=

        $=Pr(Y_i \backslash Y_1)* \ldots * Pr(Y_i \backslash Y_N)$
      - set Prop(i) = $w(i)/?_i\, w(i)$
- continue DO loop over activities

2.3 CALL subroutine SUBNET to update cost allocation matrix with cost allocation proportions.

- find subnetworks of all key events
- remove activities from subnetworks of key event i that are members of the subnetwork of key event j, where j : i
- DO over activities
  - IF common activity THEN
    - update matrix with cost allocation proportions

3. INITIALIZE VARIABLES FOR MONTE CARLO SIMULATION

3.1 CALL XEXPEC to determine mean and std.dev. of activities (pre-programmed for allowed density functions) for use in setting the limits of the empirical distributions.

3.2 CALL INITIALIZE
- -determine mean of cost distributions
- -DO over key events
  - -set endpoints of duration dist.'s to be ~ 3 std.dev.'s from mean
  - -allow user to change endpoints
  - -obtain number of cells in distributions from user

  - -set endpoints of cost dist.'s to be ~ a distance from the mean by some heuristic
  - -allow user to change endpoints
  - -obtain number of cells in distributions from user

- -obtain same info for TOTAL COST distribution

3.3 CALL SAMPLESIZE to determine required sample size for MCS
- -present menu to either get sample size info. or proceed with calc.'s
- -obtain maximum difference between sample df and true df from user
- -obtain confindence level from user
- -determine sample size via Kolmogorov-Smirnov statistic
- -allow user to change sample size if desired

4. PERFORM MONTE CARLO SIMULATION

- -DO over number of Monte Carlo samples
  - -CALL RNGENERATOR
    - -DO over activities
      - -generate activity duration
      - -generate activity cost
  - -CALL CRITICAL
    - -determine critical path
      - -calculate early start & early finish times
      - -calculate activity floats
    - -DO over activities
      - -IF activity is on critical path THEN
        - -increment critical indices counter
    - -determine realization times of all key events
  - -CALL PAYMENTS
    - -determine cash "outflow" for network
      - -DO over activities
      - -DO over periods
        - -IF activity is "active" in period THEN
          - -increment array for MCS
      - -continue over periods and activities
      - -add in any fixed costs that activities have
    - -alter cash "outflow" in accordance with inflation rate
    - -determine cost of subgraphs of key events
    - -determine total project cost
  - -CALL CDF
    - -increment empirical distributions arrays of duration and cost of key events and total project cost
    - -increment mean and variance variables for duration and cost of key events and total project cost (using standard computa-

tional formulas for mean and variance)

- CALL SMCORR
  - update correlation variables and regression variables (correlation and linear least squares regression between cost and duration of key events and total cost vs. proj. dur.)
    - DO over key events
      - update sum and sum of squares of duration and cost
    - update sum and sum of squares of total project cost
- CALL INCOME
  - update variables keeping track of the fractions of the total project cost that the key events comprise
    - DO over key events
      - remove fraction of cost to be received at proj. start
      - determine fraction of cost of subgraph of key event
      - update array containing fractions of key events over MCS
- CALL UPPAY
  - update cash "inflow" for project for each key event
    - DO over key events
      - determine the period that the key event was realized
      - update variables for "inflow" for MCS

Continue over DO loop

- CALL DISTMV
  - make final calculations on empirical distributions, means and variances of key events, and criticality indices of activities for MCS
- CALL KEINFO
  - output to disk file empirical distribution information and criticality indices
- CALL CORRF
  - make final calculations on sample correlation coefficients and least squares regression coefficients for MCS

5. OUTPUT <u>ALL</u> PERTINENT DATA TO FILE cashflow.dat FOR USE BY PROGRAM CASH

6. PRINT TERMINATION INFORMATION AND STOP PROGRAM EXECUTION.

Appendix 9.3. Program Listing of AN-COST

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                  C
C    PROGRAM   A N - C O S T               DATE:   28 OCT  86      C
C                                                                  C
C    WRITTEN   BY RUSSELL   S. VOGTMANN                            C
C                                                                  C
C    THIS PROGRAM  SIMULATES  AN ACYCLIC  NETWORK  BY MONTE  CARLO  C
C    TECHNIQUE  TO DETERMINE  THE EMPIRICAL  DENSITY  FUNCTIONS  OF C
C    REALIZATION  TIME AND COST AT VARIOUS KEY EVENTS  IN THE      C
C    NETWORK  AND EXPECTED  CASH FLOW STREAMS.                     C
C                                                                  C
C    FOR MORE DETAILS,  REFER  TO MASTER'S  THESIS  BY RUSSELL  S. C
C    VOGTMANN.  MORE DETAILS ARE ALSO GIVEN IN THE PROGRAM  UNDER  C
C    PROGRAM  INFORMATION.                                         C
C                                                                  C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C        VARIABLES:
C
C   ACOST(   ):   Contains sampled values of activity cost (does not include
C                 fixed cost).
C   ALPHA:        Interest rate on money deposited.
C   ASCEND(   ):  List of nodes in ascending order.
C   AVERG(   ):   Array containing the means of the key events obtained from
C                 file TRNS.ANC after running DODIN's program.
C   BETA:         Interest reate on money borrowed.
C   BHATO(   ):   Intercept value in least squares regression on duration
C                 and cost of key events.
C   BHAT1(   ):   Slope value in least squares regression on duration and
C                 cost of key events.
C   CORR(   ):    Sample correlation of cost and duration of key events.
C   DECEND(   ):  List of nodes in descending order.
C   DESIRE:       Desired percentage of profit.
C   DF(  ,  ):    Array of the observed density functions of the key events.
C   DIST(  ,  ):  Array containing the probabilities associated with durations
C                 of discrete duration densities.
C   DURA(  ,  ):  Array containing the durations of discrete duration
C                 densities.
C   DURATN(   ):  Contains the sampled values of activity durations.
C   EF(I):        The early finish time of activity I for a particular MCS.
C   ES(I):        The early start time of activity I for a particular MCS.
C   FRAC(   ):    The fraction of project cost that is attributed to each key
C                 event for one MC sample (this is before the percentage is
C                 taken out for what we want to receive at time C).
C   FRACTN(   ):  The fraction of project cost that is attributed to each key
C                 event for the entire simulation (this is after the percentage
C                 is taken out for, STPERC, the percentage that we want at
C                 time 0).
C   KEYDUR(   ):  Array that holds the observed duration to key events 1,n-1
C                 for each MCS.
C   KEYEVN(   ):  Array containing the numbers of the key events.
```

```
C    IWIDTH(    ):   Array  containing   the  interval   width  of  the  empirical   density
C                    functions   for  the  key  events  and  project   cost.
C    LL(  ),RR(   ):  Left  and  Right  endpoints   of  the  distributions   of  time  to
C                      the  key  events.
C    MEAN(   ):      Array  containing   the  sample   means  of  the  key  events   and  the
C                    ke  event   cost.
C    NOP(   ):       Number  of  pairs  of  duration   and  probability   for  discrete
C                    activity   durations.
C    NKEYEN:         Number  of  key  events.
C    NODEFI(I):      The  node  that  an  activity   procedes   to.
C    NODEST(I):      The  node  that  an  activity   emanates   from.
C    PAY(  ,  ):     Array  (nkeyen,mxperd)    containing   the  cost  of  subgraph   for
C                    each  key  event  during  which  period  it  was  realized   in.
C    RATE:           Retention   Percentage.
C    STDV(   ):      Array  containing   the  standard   deviation's   of  the  key  events
C                    obtained  from  file  TRNS.ANC   after  running   DODIN's   program.
C    STPERC:         Percentage   of  project   cost  that  we  want  to  receive   at  time  0.
C    SUBCO(   ):
C    SUBM(  ,  ):    The  matrix   that  contains   the  cost  associated   from  an  activity
C                    to  each  key  event.
C    SUBN(  ,  ):    A  dummy  matrix   used  in  conjunction   with  SUBM.
C    SSUM(   ):      Contains   the  sum  of  the  durations   and  costs  of  the  key  events
C                    to  be  used  in  calculation   of  correlation/least     squares   info.
C    SSUMSQ(    ):   Contains   the  sum  of  the  squares   of  duration   and  cost  of
C                    key  events.
C    SUMXY(   ):     Contains   the  sum  of  the  duration*cost    of  key  events.
C    TDURAT:         Var.  of  duration   of  node  n  (terminal   node).
C    TITLE:          Title  of  the  project.
C    VAR(  ):        Array  of  sample   variances   of  key  events  and  tot.  project   cost.
C    XEXP(   ):      Array  containing   the  expected   value's   of  the  activity   duration.
C    XSTD(   ):      Array  containing   the  standard   dev.'s   of  the  activity   duration.
C
C
C    CDUM(   ):      The  number   of  the  particular   density   of  the  duration   of
C                    an  activity.
C
C    P1(  ),P2(   ),P3(   ),P4(   ):   Parameters   of  the  density   functions   of
C                                     activity   durations.
C
C               DNUM     DENSITY          P1            P2            P3          P4
C               ---------------------------------------------------------
C                1       UNIF.CON.        --            --            L.END       R.END
C                2       TRIANG.          MODE          --            L.END       R.END
C                3       NORMAL           MEAN          VAR.          L.END       R.END
C                4       EXPONENTIAL      MEAN          --            --          --
C                5       GAMMA            ALPHA1        ALPHA2        --          --
C                6       BETA             SHAPE1        SHAPE2        L.END       R.END
C                7       DISCRETE                  <<<ORDERED    PAIRS>>>
C              ---------------------------------------------------------
C
C
C   CPDF(   ):       Character   array  containing   the  name  of  the  act.  cost  density.
C   CDNUM(   ):      The  number   of  the  particular   density   of  activity   cost.
```

```
C
C  CP1(   ):        Fixed   cost   associated    with   an  activity.
C  CP2(   ),CP3(   ),CP4(   ),CP5(   ):   Parameters    of  the   cost   functions.
C
C
C             CDNUM    DENSITY            CP2                  CP3           CP4     CP5
C             -----------------------------------------------------
C               1      CONST*Y            CONSTANT             ---           ---
C               2      NORMAL             MEAN=CP2*DUR.        VAR.        L.LIM    R.LIM
C               3      UNIFORM            SEE   THESIS
C               4      TRIANGULAR         MODE=CP2*DUR.        ---         L.LIM    R.LIM
C             -----------------------------------------------------
C
C  --->    FOR  MORE   INFORMATION    ON  COST   FUNCTIONS,    SEE  MASTER'S    THESIS
C          BY  RUSSELL    S.  VOGTMANN    "PROJECT    BIDDING    UNDER   CHANCE    TIME
C          ESTIMATES"
C
C   MAIN+
C          SUBROUTINES:
C
C   CDF:  ------------       UPDATES   EMPIRICAL    DIST.'S    IN  MONTE   CARLO   SIM.
C   CRITICAL:    -------     DETERMINES    CRITICAL    PATH   IN  MONTE   CARLO   SIM.
C   DISTMV:   ---------      DETERMINES    FINAL   VALUES   FOR  EMPIRICAL    DIST.'S   AND
C                            MEAN'S   AND  VAR.'S   AFTER   COMPLETION    OF  M.C.   SIM.
C   INCOME:   ---------      UPDATE   VARIABLE    FRACTN   THAT   KEEPS   TRACK   OF  THE
C                            FRACTION    OF  THE   TOTAL   PROJECT   COST   THAT   A  SUBGRAPH
C                            OF  A  KEY   EVENT   IS.
C   INITIALIZE:     -----    DETERMINES    REQUIRED    VALUES    FOR  EMPIRICAL    DIST.'S
C   INTRO:   ----------      PRINTS   USER   INFO
C   KEINFO:   ---------      OUTPUTS   DATA   ON  EMPIRICAL    DIST.'S    (REAL.TIME    AND   COST)
C   PAYMENTS:    -------     DETERMINES    CASH   "OUTFLOW"    AND  COST   OF  SUBGRAPH'S
C
C   RNGENERATOR:      ----   OBTAINS   RANDOM   VALUES   OF  DURATION    AND  COST   DURING
C                            MONTE   CARLO   SIM.   FOR  EACH   ACTIVITY    IN  NETWORK.    THE
C                            FOLLOWING    SUBROUTINES    ARE  CALLED   FROM   RNGENERATOR
C       --BETA    -------    GENERATES    DURATION    FROM   BETA   DISTRIBUTION
C       --EXPON    ------    GENERATES    DURATION    FROM   EXPONENTIAL     DISTRIBUTION
C       --GAMMA    ------    GENERATES    DURATION    FROM   GENERAL    GAMMA    DISTRIBUTION
C       --RANDOM    -----    GENERATES    UNIFORM(0,1)     VALUE
C       --RCOST    ------    GENERATES    COST   FOR  EACH   ACTIVITY    IN  NETWORK    ONCE   GIVEN
C                            THE  ACTIVITIES     DURATION
C       --RNORMAL    ----    GENERATES    VALUE   (DURATION    OR  COST)    FROM   NORMAL    DIST.
C       --TRIANGULAR     -   GENERATES    VALUE   (DURATION    OR  COST)    FROM   TRIANGULAR     DIST.
C       --UNICON    -----    GENERATES    VALUE   (DURATION    OR  COST)    FROM   UNIFORM    DIST.
C
C   SEPARATE:    -------     SEPARATES    ACTIVITY    COST   FOR  COMMON    ACTIVITIES     IN  NETWORK
C   UPPAY:   ----------      UPDATES   THE  ARRAY   PAY(   ,  )  THAT   TRACKS   THE  CASH
C                            "INFLOW"    OF  THE  PROJECT    (PER   PERIOD)
C   XEXPEC:   ---------      DETERMINES    VALUES   OF  EXPECTED    VALUE   AND  STD.DEV.    FOR
C                            ACTIVITIES     IN  NETWORK
C
$LARGE
      INTEGER       ASCEND(100),DECEND(100),CYCLE,DNUM(100),LL(22),RR(22),
```

```
     *             CDNUM(100),FLAGONE,FLAG2,FLAG3,CMODEL,NOP(100),
     *             NCELLS(22),NODEST(100),NODEFI(100),KEYEVN(10)
      REAL         LS(100),LSA(100),LF(100),DN(100),XEXP(100),XSTD(100),
     *             CI(100),KEYDUR(10),IWIDTH(22),DF(22,102),SDISBUR(100),
     *             PAY(10,100),PERCEN(100,10),ALPHA,BETA,RATE,POWER,
     *             TWORTH,STPERC,DIST(100,30),DURA(100,30),P1(100),P2(100)
     *             ,P3(100),P4(100),CP1(100),CP2(100),CP3(100),CP4(100),
     *             CP5(100),AVERG(10),STDV(10),DURATN(100),ACOST(100),
     *             ES(100),EF(100),EFA(100),TF(100),FF(100),SF(100),
     *             SUBN(100,10),SUBM(100,10),LP(100)
      DOUBLE   PRECISION     MEAN(22),VAR(22),SUM2(22),SSUM(21),SSUMSQ(21),
     *                       SUMXY(11),CORR(11),BHAT0(11),BHAT1(11),
     *                       FRACTN(10),PCOST,SUBCO(11)
      REAL*8   IX
      CHARACTER*6     FNAME,OUTP
C
      CALL   INTRO
C
    2 FORMAT(25(/))
    3 FORMAT(5(/))
C
C---OPEN    FILE   FOR   INPUT   OF   NETWORK    AND   ACTIVITY    DURATION/COST      INFO.
C
      WRITE(*,*)     'TYPE    FILE    NAME    CONTAINING     DURATION/COST      DATA'
      READ(*,4)     FNAME
    4 FORMAT(A6)
      OPEN   (UNIT=1,FILE      =FNAME,STATUS      ='OLD')
C
C---OPEN    FILE   FOR   OUTPUT   OF   GENERAL    INFO.
C
      WRITE(*,3)
      WRITE(*,*)     'TYPE    FILE    NAME    TO   CONTAIN    GENERAL    OUTPUT'
      READ(*,4)    OUTP
      OPEN   (UNIT=3,FILE      =OUTP,STATUS      ='NEW')
C--OPEN    FILE   "KEDIST.DAT"      FOR   OUTPUT    OF   REAL.   TIME   DIST.'S
      OPEN   (UNIT=4,FILE      = 'KEDIST.DAT'      ,FORM    = 'FORMATTED',
     *                 ACCESS    = 'SEQUENTIAL',      STATUS    = 'NEW')
C--OPEN    FILE   "CASHFLOW.DAT"       FOR   TESTING    ONLY
      OPEN   (UNIT=8,FILE      = 'CASHFLOW.DAT'      ,FORM    = 'FORMATTED',
     *                 ACCESS    = 'SEQUENTIAL',      STATUS    = 'NEW')
C--OPEN    FILE   "COSTD.DAT"      FOR   OUTPUT    OF   COST    DIST.'S
      OPEN   (UNIT=7,FILE      = 'COSTD.DAT'      ,FORM    = 'FORMATTED',
     *                 ACCESS    = 'SEQUENTIAL',      STATUS    = 'NEW')
C--OPEN    FILE   "TRNS.ANC"     FOR   INPUT   OF  MEAN'S    AND   STD.   DEV.'S    OF
C    KEY   EVENTS    DETERMINED      FROM   DOD1TRNS.EXE
      OPEN   (UNIT=9,FILE='TRNS.ANC',STATUS='OLD')
C
C---VARIABLE      INITIALIZATION
C
      DO  5  I=1,100
        CI(I)=0.0
        NOP(I)=0
        XEXP(I)=0.0
```

```
           XSTD(I)=0.0
           SDISBUR(I)=0.0
    5   CONTINUE
C
        DO  6  I=1,22
        DO  7  J=1,102
           MEAN(I)=0.0
           VAR(I)=0.0
           SUM2(I)=0.0
           DF(I,J)=0.0
    7   CONTINUE
    6   CONTINUE
C
        DO  8  I=1,10
        DO  9  J=1,100
           PAY(I,J)=0.0
           FRACTN(I)=0.0
    9   CONTINUE
    8   CONTINUE
C
        DO  1  I=1,11
           SUMXY(I)=0.0
           BHAT0(I)=0.0
           BHAT1(I)=0.0
           CORR(I)=0.0
    1   CONTINUE
C
        DO  19  I=1,21
           SSUM(I)=0.0
           SSUMSQ(I)=0.0
   19   CONTINUE
C-----------------------------------------------------------------
C       READ   DATA   FROM   INPUT   FILE
C-----------------------------------------------------------------
        READ(1,11)     SCAL,NODES,NACT,NRR,NCONT,MCS,NSIM,NKEYEN,NANG
   11   FORMAT(F5.3,10I5)
        READ   (1,106)    (  KEYEVN(I),      I=1,NKEYEN      )
  106   FORMAT(14I5)
        DO  10  I=1,NACT
           READ(1,103)      NODEST(I),NODEFI(I),DNUM(I),NOP(I)
           IF  (DNUM(I).EQ.7)       THEN
              READ(1,104)      (DURA(I,J),DIST(I,J),J=1,NOP(I))
           ELSE
              READ(1,104)      P1(I),P2(I),P3(I),P4(I)
           ENDIF
   10   CONTINUE
C
        DO  90  I=1,NACT
           READ(1,103)      CDNUM(I)
           READ(1,104)      CP1(I),CP2(I),CP3(I),CP4(I),CP5(I)
   90   CONTINUE
C
  103   FORMAT(14I5)
```

```
  104  FORMAT(6E11.4)
C
       WRITE(*,2)
       WRITE(*,22)
   22  FORMAT(5X,'YOU      WILL   NOW   BE  ASKED   TO  INPUT   VALUES   THAT   WILL   BE',/,
     *5X,'USED    IN  THE   COST   CALCULATIONS')
       WRITE(*,3)
       WRITE(*,23)
   23  FORMAT(2X,'INPUT      PERIOD    LENGTH   (i.e.   5 days,   etc.   --THIS    MUST   BE  I
     *NTEGER    VALUED)',//)
       READ(*,*)     CYCLE
       WRITE(3,13)      CYCLE
   13  FORMAT(/1X,'THE       PERIOD    LENGTH    =  ',I5)
  111  WRITE(*,3)
       WRITE(*,12)
       READ(*,*)     STPERC
       IF  (STPERC.GT.100)       THEN
         WRITE(*,*)      'OUT   OF   RANGE   --  TRY   AGAIN'
         GOTO    111
       ENDIF
       WRITE(3,14)      STPERC
   14  FORMAT(/1X,'THE       PERC.   REQUIRED    AT   START    =  ',F6.3)
       STPERC=STPERC/100
   12  FORMAT(1X,'INPUT       PERCENTAGE     OF   COST   TO  RECEIVE    AT   PROJECT    START')
  112  WRITE(*,3)
       WRITE(*,*)      'INPUT    THE   INFLATION     RATE   (NOT   A  PERCENTAGE)'
       READ(*,*)     GAMMA
       IF  (GAMMA.GT.1)      THEN
         WRITE(*,*)      'OUT   OF   RANGE   --  TRY   AGAIN'
         GOTO    112
       ENDIF
       WRITE(3,16)     GAMMA
   16  FORMAT(/1X,'THE       INFLATION     RATE   =  ',F6.5)
       WRITE(*,3)
       WRITE(*,*)      'INPUT    RANDOM    NUMBER    SEED'
       READ(*,*)     IX
       WRITE(3,17)      IX
   17  FORMAT(/1X,'THE       INITIAL    RANDOM    NUMBER    SEED    =  ',F20.2)
C
C---INPUT     MEAN   AND   VARIANCE    VALUE'S    FROM    FILE    TRNS.ANC
C
       DO  92  I=1,NKEYEN
         READ(9,104)     AVERG(I),STDV(I)
 92    CONTINUE
C-----------------------------------------------------------------
C      DETERMINE    THE   NODES   IN  ASCENDING    ORDER   FOR   USE   CRITICAL    PATH
C      CALCULATIONS
C-----------------------------------------------------------------
       ASCEND(1)     =  NODEST(1)
       LDT   =1
       DO  210   I  =  1,NACT
         KDT   =  LDT
         DO  200   K  =  1,KDT
```

```
          IF  ( NODEST(I)    .EQ.   ASCEND(K)    ) GO  TO  155
  200     CONTINUE
          LDT   = LDT   + 1
          ASCEND(LDT)     = NODEST(I)
  155     DO  205  K  =  1,KDT
            IF  ( NODEFI(I)    .EQ.   ASCEND(K)    ) GO  TO  210
  205     CONTINUE
          LDT   = LDT   + 1
          ASCEND(LDT)     = NODEFI(I)
  210   CONTINUE
        LM  = LDT   - 1
        DO  215  I  =  1,LM
          M  =  I  +  1
          DO  215  K  =  M,LDT
          IF  ( ASCEND(I).LE.ASCEND(K)        ) GO  TO  215
          KTEMP   = ASCEND(I)
          ASCEND(I)     = ASCEND(K)
          ASCEND(K)     = KTEMP
  215   CONTINUE
C-----------------------------------------------------------------
C       NODES  IN  DESCENDING   ORDER   FOR  SAME  PURPOSE   AS  ABOVE
C-----------------------------------------------------------------
        DO  220  I  =  1,LDT
          DECEND(I)     = ASCEND(I)
  220   CONTINUE
        DO  225  J  =  1,LM
          L  =  J  +  1
        DO  225  K  =  L,LDT
          IF  ( DECEND(J).GE.DECEND(K)        ) GO  TO  225
          KTEMP   = DECEND(J)
          DECEND(J)     = DECEND(K)
          DECEND(K)     = KTEMP
  225   CONTINUE
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C    THE  FOLLOWING   PROGRAM   SECTION   OBTAINS   THE  PERCENTAGES    TO  BE
C    USED  IN  SPLITTING   THE  COSTS   OF  THE  COMMON   ACTIVITIES.
C
C    HERE,  I  ASSIGN  A  COST  OF  1  TO  ALL  ACTIVITIES   TO  GET  THE  MATRIX
C    THAT  ASSIGNS  THE  COST  OF  THE  ACTIVITIES   TO  THE  VARIOUS   KEY  EVENTS.
C
        DO  93  I=1,NACT
          ACOST(I)=1
   93   CONTINUE
C
C    1.  THE  FIRST  CALL  TO  SUBNET  DETERMINES   THE  SUBNETWORKS   ASSOCIATED
C        WITH  ALL  THE  KEY  EVENTS.
C    2.  THE  CALL  TO  SEPARATE  OBTAINS  THE  PERCENTAGES   OF  THE  COMMON  ACT.'S
C        TO  ASSIGN  TO  ITS  KEY  EVENTS.
C    3.  THE  SECOND  CALL  TO  SUBNET  THEN  DETERMINES   THE  MATRIX  OF  PERCEN-
C        TAGES  THAT  ASSIGNS  THE  COSTS  OF  ACTIVITIES   TO  KEY  EVENTS.
C
        CALL  SUBNET(NKEYEN,NACT,KEYEVN,NODEFI,NODEST,ACOST,ASCEND,SUBCO,
```

```
     *              SUBM,PERCEN,FLAGONE)
      FLAGONE=1
      CALL   SEPARATE(SUBM,I,NKEYEN,NODEST,NODEFI,KEYEVN,AVERG,STDV,
     *                 PERCEN,NACT,ACOST)
      CALL   SUBNET(NKEYEN,NACT,KEYEVN,NODEFI,NODEST,ACOST,ASCEND,SUBCO,
     *              SUBM,PERCEN,FLAGONE)
C
      CALL   XEXPEC(XEXP,XSTD,NACT,P1,P2,P3,P4,DNUM,DIST,DURA,NOP)
C
      CALL   INITIALIZE(NKEYEN,LL,RR,IWIDTH,NMCS,KEYEVN,FLAGONE,AVERG,
     *                   STDV,SUBM,CP1,CP2,NACT,XEXP,XSTD,OUTP,NCELLS)
C
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     MONTE   CARLO   SIMULATION    OF  NETWORK
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      MXPERD=0
C
      WRITE(*,3)
      WRITE(*,*)    'BEGIN   MONTE   CARLO   SAMPLING'
      DO  500  ICOUNT=1,NMCS
         IF  (MOD(ICOUNT,50).EQ.0)        WRITE(*,*)    ICOUNT
C
         CALL   RNGENERATOR(NACT,DURATN,DNUM,P1,P2,P3,P4,ACOST,CDNUM,
     *                    CP1,CP2,CP3,CP4,CP5,DIST,DURA,NOP,IX)
         CALL   CRITICAL(NACT,NODEST,NODEFI,DURATN,DECEND,ASCEND,LM,LDT,
     *                  NKEYEN,KEYEVN,KEYDUR,CI,ES,EF)
         CALL   PAYMENTS(KEYDUR,NACT,CYCLE,ACOST,DURATN,ES,EF,SDISBUR,
     *                  NMCS,MXPERD,CP1,GAMMA,SUBCO,SUBM,NKEYEN,PCOST)
         CALL   CDF(LL,RR,IWIDTH,KEYDUR,DF,NKEYEN,MEAN,SUM2,SUBCO,PCOST)
         CALL   SMCORR(PCOST,SUBCO,KEYDUR,NKEYEN,SSUM,SSUMSQ,SUMXY)
         CALL   INCOME(NKEYEN,SUBCO,PCOST,STPERC,FRACTN,NMCS)
         CALL   UPPAY(KEYDUR,PAY,MXPERD,CYCLE,NMCS,NKEYEN,SUBCO,CAPITL)
C
  500 CONTINUE
      WRITE(*,*)    'MONTE   CARLO   SAMPLING    COMPLETE'
C
C---THE   NEXT   TWO   CALLS   (DISTMV   & KEINFO)   FINISHES   THE   CALCULA-
C---TIONS    ON  THE   ARRAY   DF  AND   ON  THE  MEAN,   VAR.,   AND  CI,  THEN   SENDS
C---THE    INFO.   TO  AN  EXTERNAL   DATA   FILE.
C
      CALL   DISTMV(DF,MEAN,SUM2,VAR,NKEYEN,NMCS,CI,NACT)
      CALL   KEINFO(DF,MEAN,VAR,NKEYEN,LL,RR,IWIDTH,CI,NACT,KEYEVN)
C
C---THE   NEXT   CALL   TO  CORRF   FINISHES   THE  CALCULATIONS    ON  THE   SAMPLE
C---CORRELATIONS      BETWEEN    KEY   EVENT   DURATION    AND   COST,   AS  WELL   AS  THE
C---CALCULATIONS      ON  THE   LEAST   SQUARES    FIT.
C---L.S.    EQUATION    ==>   COST   =  BHATO   +  BHAT1*DURATION
      CALL   CORRF(NKEYEN,NMCS,SSUM,SSUMSQ,SUMXY,CORR,BHATO,BHAT1)
C
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     MONTE   CARLO   SIMULATION    ENDS
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
```

```
C---OUTPUTTING CASH FLOW AND OTHER PERTINENT VALUES TO FILE
C    "CASHFLOW.DAT" FOR LATER USE
C
      WRITE(8,650)   CYCLE
      WRITE(8,650)   MXPERD
      WRITE(8,651)   (SDISBUR(I),I=1,MXPERD)
      WRITE(8,650)   NKEYEN
      WRITE(8,650)   (KEYEVN(I),I=1,NKEYEN)
      DO 501 J=1,NKEYEN
        WRITE(8,651)   (PAY(J,I),I=1,MXPERD)
  501 CONTINUE
      WRITE(8,651)   STPERC,CAPITL
      DO 502 J=1,NKEYEN
        WRITE(8,651)   MEAN(J),SQRT(VAR(J))
        WRITE(8,651)   MEAN(J+10),SQRT(VAR(J+10))
  502 CONTINUE
      WRITE(8,651)   MEAN(22),SQRT(VAR(22))
  650 FORMAT(10I5)
  651 FORMAT(5E13.7)
C
      DO 660 I=1,NKEYEN
        WRITE(8,690)   LL(I),RR(I),NCELLS(I),IWIDTH(I)
        WRITE(8,651)   (DF(I,J),J=1,NCELLS(I)+2)
        WRITE(8,690)   LL(I+10),RR(I+10),NCELLS(I+10),IWIDTH(I+10)
        WRITE(8,651)   (DF(I+10,J),J=1,NCELLS(I+10)+2)
  660 CONTINUE
      WRITE(8,690)   LL(22),RR(22),NCELLS(22),IWIDTH(22)
      WRITE(8,651)   (DF(22,J),J=1,NCELLS(22)+2)
  690 FORMAT(I10,I10,I10,F15.5)
C
      DO 665 I=1,NKEYEN
        WRITE(8,651)   CORR(I),BHAT0(I),BHAT1(I)
  665 CONTINUE
      WRITE(8,651)   CORR(11),BHAT0(11),BHAT1(11)
C
      DO 666 I=1,NKEYEN
        WRITE(8,651)   FRACTN(I)
  666 CONTINUE
C
C---TERMINATION COMMENTS TO SCREEN
C
      WRITE(*,2)
      WRITE(*,653)
  653 FORMAT(/1X,'PROGRAM AN-COST.EXE IS TERMINATING ---',////,
     *10X,'1.  OUTPUT INFORMATION',/,
     *10X,'2.  EXIT PROGRAM',//,
     *5X,'ENTER CHOICE',//)
      READ(*,*)   ICHOICE
      IF (ICHOICE.LT.1.OR.ICHOICE.GT.2)   THEN
        WRITE(*,*)   'OUT OF RANGE --- EXITING PROGRAM'
        GOTO 655
      ENDIF
      IF (ICHOICE.EQ.1)   CALL OUTHELP
```

```
C
  655  STOP
       END
C------------------------------------------------------------------
C
C        SUBROUTINE      C D F
C
C            THIS  SUBROUTINE   INCREMENTS   THE  DIST.  FUNC.  ARRAYS  FOR
C        THE  KEY  EVENTS  AND  PROJECT  COST.   IT  ALSO  TRACKS  VALUES  TO
C        DETERMINE  THE  MEAN  AND  VARIANCE  OF  THE  REALIZATION  TIME  OF
C        THE  KEY  EVENTS,  AS  WELL  AS  FOR  THE  PROJECT  COST.
C
C------------------------------------------------------------------
C
       SUBROUTINE   CDF  (LL,RR,IWIDTH,KEYDUR,DF,NKEYEN,MEAN,SUM2,
     *                     SUBCO,PCOST)
C
       INTEGER    LL(22),RR(22),NKEYEN
       REAL       IWIDTH(22),KEYDUR(10),DF(22,102)
       DOUBLE  PRECISION   MEAN(22),SUM2(22),PCOST,SUBCO(11)
C
C---THIS  <DO>  LOOP  INCREMENTS  THE  ARRAY  DF  FOR  ALL  KEY  EVENTS.
C
       DO  7000  K=1,NKEYEN
         XX=(KEYDUR(K)-LL(K))/IWIDTH(K)
         IF  (XX-INT(XX).EQ.0.0)       THEN
           MARK=INT(XX)+1
         ELSE
           MARK=INT(XX)+2
         ENDIF
         IF  (MARK.LE.0)     MARK=1
         IF  (KEYDUR(K).GT.RR(K))       MARK=INT((RR(K)-LL(K))/IWIDTH(K))+2
         DF(K,MARK)=DF(K,MARK)+1
 7000  CONTINUE
C
C---NOW   INCREMENT   FOR  COST  OF  SUBGRAPHS   OF  KEY  EVENTS
C
       DO  7020  K=11,NKEYEN+10
         XX=(SUBCO(K-10)-LL(K))/IWIDTH(K)
         IF  (XX-INT(XX).EQ.0.0)       THEN
           MARK=INT(XX)+1
         ELSE
           MARK=INT(XX)+2
         ENDIF
         IF  (MARK.LE.0)     MARK=1
         IF  (SUBCO(K-10).GT.RR(K))       MARK=INT((RR(K)-LL(K))/IWIDTH(K))+2
         DF(K,MARK)=DF(K,MARK)+1
 7020  CONTINUE
C
C---NOW   FOR  THE  TOTAL  PROJECT   COST
C
       XX=(PCOST-LL(22))/IWIDTH(22)
       IF  (XX-INT(XX).EQ.0.0)       THEN
```

```
          MARK=INT(XX)+1
        ELSE
          MARK=INT(XX)+2
        ENDIF
        IF  (MARK.LE.0)    MARK=1
        IF  (PCOST.GT.RR(22))       THEN
          MARK=INT((RR(22)-LL(22))/IWIDTH(22))+2
        ENDIF
        DF(22,MARK)=DF(22,MARK)+1
C
C----UPDATING     MEAN   AND   VARIANCE    COUNTERS    ON  DURATION    AND   COST
C
        DO  7005   K=1,NKEYEN
          MEAN(K)=MEAN(K)+KEYDUR(K)
          SUM2(K)=SUM2(K)+KEYDUR(K)*KEYDUR(K)
          MEAN(K+10)=MEAN(K+10)+SUBCO(K)
          SUM2(K+10)=SUM2(K+10)+SUBCO(K)*SUBCO(K)
 7005   CONTINUE
C
C---UPDATE    MEAN   AND   VAR.   COUNTERS    FOR   PROJECT    COST
C
        MEAN(22)=MEAN(22)+PCOST
        SUM2(22)=SUM2(22)+PCOST*PCOST
C
        RETURN
        END
C-----------------------------------------------------------------------
C
C       SUBROUTINE      C O R R F
C
C       THIS  SUBROUTINE    COMPLETES    THE  CALCULATIONS     FOR  SAMPLE    CORRELATION
C       AND  THE  LEAST   SQUARES    ESTIMATORS.
C
C-----------------------------------------------------------------------
C
        SUBROUTINE     CORRF(NKEYEN,NMCS,SSUM,SSUMSQ,SUMXY,CORR,BHATO,BHAT1)
C
        INTEGER    NKEYEN,NMCS
        DOUBLE    PRECISION     SSUM(21),SSUMSQ(21),SUMXY(11),RNUM,DEN,CORR(11),
     *                          BHATO(11),BHAT1(11)
C
C---DETERMINE     SAMPLE    CORRELATION
C
        RNUM=0.
        DEN=0.
        DO  3700   I=1,NKEYEN
          RNUM=NMCS*SUMXY(I)-SSUM(I)*SSUM(I+10)
          DEN=(NMCS*SSUMSQ(I)-SSUM(I)**2)*(NMCS*SSUMSQ(I+10)-SSUM(I+10)**2)
          IF  (DEN.EQ.0.)      THEN
            CORR(I)=-9999
          ELSE
            CORR(I)=RNUM/SQRT(DEN)
          ENDIF
```

```
 3700   CONTINUE
C
        RNUM=NMCS*SUMXY(11)-SSUM(NKEYEN)*SSUM(21)
        DEN=(NMCS*SSUMSQ(NKEYEN)-SSUM(NKEYEN)**2)*(NMCS*SSUMSQ(21)-SSUM(21
       *)**2)
C
C   IF  DEN  EQUALS   ZERO,   THE   NETWORK   IS  DETERMINISTIC    AND   CORR   MAKES   NO  SENSE
C
        IF  (DEN.EQ.0.)     THEN
          CORR(11)=-9999
        ELSE
          CORR(11)=RNUM/SQRT(DEN)
        ENDIF
C
C---DETERMINE      THE   ESTIMATORS    OF  THE   LEAST    SQUARES    FIT
C
        DO  3710   I=1,NKEYEN
          RNUM=NMCS*SUMXY(I)-SSUM(I)*SSUM(I+10)
          DEN=NMCS*SSUMSQ(I)-SSUM(I)**2
          IF  (DEN.EQ.0.)     THEN
            BHAT1(I)=-9999
          ELSE
            BHAT1(I)=RNUM/DEN
          ENDIF
C
          RNUM=SSUMSQ(I)*SSUM(I+10)-SSUM(I)*SUMXY(I)
          IF  (DEN.EQ.0.)     THEN
            BHAT0(I)=-9999
          ELSE
            BHAT0(I)=RNUM/DEN
          ENDIF
 3710   CONTINUE
C
        RNUM=NMCS*SUMXY(11)-SSUM(NKEYEN)*SSUM(21)
        DEN=NMCS*SSUMSQ(NKEYEN)-SSUM(NKEYEN)**2
        IF  (DEN.EQ.0.)     THEN
          BHAT1(11)=-9999
        ELSE
          BHAT1(11)=RNUM/DEN
        ENDIF
C
        RNUM=SSUMSQ(NKEYEN)*SSUM(21)-SSUM(NKEYEN)*SUMXY(11)
        IF  (DEN.EQ.0.)     THEN
          BHAT0(11)=-9999
        ELSE
          BHAT0(11)=RNUM/DEN
        ENDIF
C
        RETURN
        END
C--------------------------------------------------------------------------
C
C       SUBROUTINE     CRITICAL
```

```
C
C         THIS  SUBROUTINE  DETERMINES  THE  CRITICAL  PATH,INCREMENTING
C      THE  CRITICAL  INDICES,  AND  DETERMINES  THE  REALIZATION  TIMES
C      OF  THE  KEY  EVENTS.
C-----------------------------------------------------------------
C
      SUBROUTINE  CRITICAL  (NACT,NODEST,NODEFI,DURATN,DECEND,ASCEND,
     *                        LM,LDT,NKEYEN,KEYEVN,KEYDUR,CI,ES,EF)
      INTEGER     NACT,NODEST(100),NODEFI(100),ASCEND(100),DECEND(100),LM,
     *            LDT,NKEYEN,KEYEVN(10)
      REAL        DURATN(100),KEYDUR(10),ES(100),LS(100),EF(100),LF(100)
     *            ,TF(100),FF(100),SF(100),EFA(100),LSA(100),CI(100)
      REAL        TDURAT
C
C
C     CALCULATE  EARLY  START  &  EARLY  FINISH
C
      DO 230  I = 1,NACT
        IF ( NODEST(I).NE.ASCEND(1)       ) GO TO 230
        ES(I)   = 0.
        EF(I)   = ES(I)  + DURATN(I)
  230 CONTINUE
C
      LM = LDT  - 1
      DO 235  J = 2,LM
        BIG  = 0.
        DO 240  I = 1,NACT
          IF ( NODEFI(I).NE.ASCEND(J)       ) GO TO 240
          IF ( EF(I).LE.   BIG  ) GO TO 240
          BIG  = EF(I)
  240   CONTINUE
        DO 245  I = 1,NACT
          IF ( NODEST(I).NE.ASCEND(J)       ) GO TO 245
          ES(I)   = BIG
          EF(I)   = ES(I)  + DURATN(I)
  245   CONTINUE
  235 CONTINUE
C
      TDURAT  = 0.0
      DO 250  I = 1,NACT
        IF ( NODEFI(I).NE.ASCEND(LDT)       ) GO TO 250
        IF ( EF(I).LE.TDURAT      ) GO TO 250
        TDURAT  = EF(I)
  250 CONTINUE
C
C---CALCULATE   LATE  START  AND  LATE  FINISH
C
      DO 255  I = 1,NACT
        IF ( NODEFI(I).NE.DECEND(1)       ) GO TO 255
        LF(I)   = TDURAT
        LS(I)   = TDURAT  - DURATN(I)
  255 CONTINUE
C
```

```
      DO 260 J = 2,LM
        SMALL  = 2147483647.
        DO 265 I = 1,NACT
          IF ( NODEST(I).NE.DECEND(J)       ) GO TO 265
          IF ( LS(I).GE.SMALL     ) GO TO 265
          SMALL  = LS(I)
  265   CONTINUE
        DO 270 I = 1,NACT
          IF ( NODEFI(I).NE.DECEND(J)       ) GO TO 270
          LF(I)  = SMALL
          LS(I)  = LF(I)  - DURATN(I)
  270   CONTINUE
  260 CONTINUE
C
C---CALCULATE    FLOATS
C
      DO 271 I = 1,NACT
        DO 266 J = 1,NACT
          IF ( NODEFI(I).NE.DECEND(1)       ) GO TO 280
          EFA(I)  = TDURAT
          GO TO 266
  280     IF ( NODEFI(I).NE.NODEST(J)       ) GO TO 266
          EFA(I)  = ES(J)
  266   CONTINUE
        DO 275 J = 1,NACT
          IF ( NODEST(I).NE.ASCEND(1)       ) GO TO 285
          LSA(I)=0
          GO TO 275
  285     IF ( NODEST(I).NE.NODEFI(J)       ) GO TO 275
          LSA(I)  = LF(J)
  275   CONTINUE
        TF(I)  = LF(I)  - ES(I)  - DURATN(I)
        FF(I)  = EFA(I)  - ES(I)  - DURATN(I)
        SF(I)  = LF(I)  - LSA(I)  - DURATN(I)
  271 CONTINUE
C
C---INCREMENT    CRITICAL    INDICES
C
      DO 290 I=1,NACT
        IF (TF(I).LT.0.00001)     CI(I)=CI(I)+1.0
  290 CONTINUE
C
C---DETERMINE    REALIZATION    TIME  OF  KEY  EVENTS
C
      DO 292 K=1,NKEYEN-1
      DO 293 I=1,NACT
        IF (KEYEVN(K).NE.NODEST(I))       GO TO 293
        KEYDUR(K)=ES(I)
        GO TO 292
  293 CONTINUE
  292 CONTINUE
C
C---SET   REAL.  TIME  OF  END-NODE   TO  TDURAT
```

```
C
      KEYDUR(NKEYEN)=TDURAT
C
      RETURN
      END
C-----------------------------------------------------------------
C
C        SUBROUTINE       D I S T M V
C
C             THIS  SUBROUTINE   NORMALIZES   THE  VALUES  IN DF AS WELL  AS
C        MAKING  THE  FINAL  CALCULATIONS   OF THE  MEAN  AND  VARIANCE   OF
C        THE  REALIZATION   TIME  OF  KEY  EVENTS.
C
C-----------------------------------------------------------------
C
      SUBROUTINE    DISTMV(DF,MEAN,SUM2,VAR,NKEYEN,NMCS,CI,NACT)
      INTEGER    NKEYEN,NMCS,NACT
      REAL       DF(22,102),CI(100)
      DOUBLE   PRECISION     MEAN(22),SUM2(22),VAR(22)
C
      DO  3000   J1=1,NKEYEN
      DO  3005   J2=1,102
        DF(J1,J2)=DF(J1,J2)/NMCS
        DF(J1+10,J2)=DF(J1+10,J2)/NMCS
 3005 CONTINUE
 3000 CONTINUE
C
      DO  3006   J2=1,102
        DF(22,J2)=DF(22,J2)/NMCS
 3006 CONTINUE
C
C---MAKE   FINAL   CALCULATIONS    ON  MEAN   AND   VARIANCE    OF  REALIZATION
C---TIMES    AND   COST   OF   KEY   EVENTS.
C
      DO  3007   J1=1,NKEYEN
        VAR(J1)=(NMCS*SUM2(J1)-MEAN(J1)*MEAN(J1))/(NMCS*NMCS-NMCS)
        MEAN(J1)=MEAN(J1)/NMCS
        VAR(J1+10)=(NMCS*SUM2(J1+10)-MEAN(J1+10)*MEAN(J1+10))/
     *                  (NMCS*NMCS-NMCS)
        MEAN(J1+10)=MEAN(J1+10)/NMCS
 3007 CONTINUE
C
C---FINAL    CALC.'S    ON  MEAN   AND   VAR.   ON  TOT.   PROJECT    COST
C
      VAR(22)=(NMCS*SUM2(22)-MEAN(22)*MEAN(22))/(NMCS*NMCS-NMCS)
      MEAN(22)=MEAN(22)/NMCS
C
      DO  3008   J1=1,NACT
        CI(J1)=CI(J1)/NMCS
 3008 CONTINUE
C
C
      RETURN
```

```
      END
C-----------------------------------------------------------------
C
C          SUBROUTINE       I  N  C  O  M  E
C
C         TAKE   OUT   THE   PERCENTAGE    OF  COST   PREDETERMINED     TO  RECEIVE    AT
C     BEGINNING    OF  PROJECT,    AND   UPDATE    OVERALL    PROJECT    COST   FRACTIONS.
C
C-----------------------------------------------------------------
C
      SUBROUTINE     INCOME(NKEYEN,SUBCO,PCOST,STPERC,FRACTN,NMCS)
C
      INTEGER    NKEYEN,NMCS
      REAL      STPERC
      DOUBLE    PRECISION     FRACTN(10),FRAC(10),SUBCO(11),PCOST
C
      DO  3510   I=1,NKEYEN
        FRAC(I)=SUBCO(I)/PCOST
 3510 CONTINUE
C
      SUBCO(NKEYEN+1)=STPERC*PCOST
      DO  3520   I=1,NKEYEN
        SUBCO(I)=FRAC(I)*(PCOST*(1-STPERC))
        FRAC(I)=SUBCO(I)/PCOST
        FRACTN(I)=FRACTN(I)+FRAC(I)/NMCS
 3520 CONTINUE
C
      RETURN
      END
C-----------------------------------------------------------------
C
C          SUBROUTINE       I  N  I  T  I  A  L  I  Z  E
C
C              THIS   SUBROUTINE    OBTAINS    THE   NECESSARY    INFO.   FOR   THE
C          DISTRIBUTION      FUNCTIONS    USED   DURING    THE   MONTE    CARLO    SIM.
C
C          --THE    DIST.    FUNCTIONS     CONTAIN    AT   MOST    ONE   HUNDRED     INTERVALS.
C
C                ---DF(--,102)      - ARRAY    CONTAINING     THE   EMPIRICAL     DIST.    FUNC.
C                ---LL(--)          - LEFT    ENDPOINT     OF  A  DIST.    FUNC.
C                ---RR(--)          - RIGHT     "         "  "     "       "
C                ---IWIDTH(--)      - INTERVAL    WIDTH    OF  THE   DIST.    FUNC.
C
C-----------------------------------------------------------------
C
      SUBROUTINE     INITIALIZE     (NKEYEN,LL,RR,IWIDTH,NMCS,KEYEVN,FLAGONE,
     *                              AVERG,STDV,SUBM,CP1,CP2,NACT,XEXP,XSTD,
     *                              OUTP,NCELLS)
C
      INTEGER      K,LL(22),RR(22),KEYEVN(10),FLAGONE,NCELLS(22)
      REAL         IWIDTH(22),AVERG(10),STDV(10),CP2(100),CP1(100),
     *             SUBM(100,10),XEXP(100),XSTD(100)
      CHARACTER*1     DCHAR
```

```
      CHARACTER*6     OUTP
      REAL   CMEAN(10),CVAR(10)
C
C---DETERMINE     SOME   LIMITS   ON  THE  COST   DISTRIBUTIONS
C
      CTOT=0.0
      CTVAR=0.0
      DO  82  I=1,NKEYEN
        CMEAN(I)=0.0
        CVAR(I)=0.0
   82 CONTINUE
C
      DO  8050   I=1,NKEYEN
      DO  8051   I1=1,NACT
        CMEAN(I)=CMEAN(I)+SUBM(I1,I)*XEXP(I1)*CP2(I1)
        CMEAN(I)=CMEAN(I)+SUBM(I1,I)*CP1(I1)
        CVAR(I)=CVAR(I)+SUBM(I1,I)*XSTD(I1)*XSTD(I1)*CP2(I1)
 8051 CONTINUE
 8050 CONTINUE
C
      DO  8054   I=1,NKEYEN
        IF  (CVAR(I).LE.0.0)       CVAR(I)=0.0
 8054 CONTINUE
C
      DO  8052   I=1,NKEYEN
        CTOT=CTOT+CMEAN(I)
        CTVAR=CTVAR+CVAR(I)
 8052 CONTINUE
      IF  (CTVAR.LE.0.0)      CTVAR=0.0
C
C---GET    INFO   ON  EMPIRICAL    DIST.'S
C
      WRITE   (*,8081)
      WRITE   (*,8081)
      WRITE   (*,8080)
      WRITE   (*,8081)
 8080 FORMAT(///1X,'THE        EMPIRICAL     DENSITY    FUNCTION'S     OF  REALIZATION',/,
     *5X,'TIME    OF  THE   KEY   EVENTS    WILL    ASSUME    THE   FOLLOWING     ENDPOINTS')
 8081 FORMAT(2(/),'        ')
C
      DO  8000   K=1,NKEYEN
        WRITE   (*,8090)    KEYEVN(K)
        LL(K)=INT(AVERG(K)-3*STDV(K))
        RR(K)=INT(AVERG(K)+3*STDV(K))
        IF  (LL(K).EQ.RR(K))       THEN
          LL(K)=LL(K)-10
          RR(K)=RR(K)+10
        ENDIF
        IF  (LL(K).LT.0.0)       LL(K)=0.0
        WRITE   (*,8091)    LL(K),RR(K)
 8090 FORMAT(//1X,'LOWER        AND   UPPER    ENDPOINTS     FOR   KEY   EVENT    ',I4,'    ARE:')
 8091 FORMAT(/10X,I10,5X,I10)
        WRITE   (*,8070)
```

```
      READ(*,8074)    DCHAR
 8070 FORMAT(/5X,'Are     These   Values   Acceptable?    (Y  or  N)')
 8074 FORMAT(A1)
      IF  (DCHAR.EQ.'N'.OR.DCHAR.EQ.'n')         THEN
        WRITE(*,8071)
 8071 FORMAT(/5X,'THEN      INPUT   YOUR   OWN   VALUES   (LOWER   END   THEN   UPPER)')
        READ(*,*)    LL(K),RR(K)
      ENDIF
C
 8010 WRITE   (*,8081)
      WRITE   (*,8093)    KEYEVN(K)
      READ   (*,*)   NCELLS(K)
      XINUM=NCELLS(K)
      IWIDTH(K)=(RR(K)-LL(K))/XINUM
 8093 FORMAT(/1X,'HOW      MANY   INVTERVALS    FOR   DIST.   OF  KEY   EVENT',I4,/,
     &3X,'--recommended       value   is  no  more   than   20',/)
      IF  (XINUM.GT.100)     THEN
         WRITE   (*,8095)
         GO  TO  8010
      END  IF
C
C---NOW    FOR   COST   DISTRIBUTIONS     OF  KEY   EVENTS
C
      WRITE   (*,8040)    KEYEVN(K)
      LL(K+10)=INT(CMEAN(K)-18*SQRT(CVAR(K)))
      RR(K+10)=INT(CMEAN(K)+18*SQRT(CVAR(K)))
      IF  (LL(K+10).EQ.RR(K+10))        THEN
        LL(K+10)=LL(K+10)-10
        RR(K+10)=RR(K+10)+10
      ENDIF
      IF  (LL(K+10).LT.0)      LL(K+10)=0
      WRITE(*,8091)       LL(K+10),RR(K+10)
 8040 FORMAT(///1X,'ENDPOINTS         FOR   COST   DIST.   OF  KEY   EVENT   ',I4,'   ARE')
      WRITE(*,8070)
      READ(*,8074)    DCHAR
      IF  (DCHAR.EQ.'N'.OR.DCHAR.EQ.'n')         THEN
        WRITE(*,8071)
        READ(*,*)     LL(K+10),RR(K+10)
      ENDIF
C
 8045 WRITE   (*,8081)
      WRITE   (*,8043)    KEYEVN(K)
      READ   (*,*)   NCELLS(K+10)
      XINUM=NCELLS(K+10)
      IWIDTH(K+10)=(RR(K+10)-LL(K+10))/XINUM
 8043 FORMAT(/1X,'HOW      MANY   INVTERVALS    FOR   COST   OF  KEY   EVENT',I4,/,
     &3X,'--recommended       value   is  no  more   than   20',/)
      IF  (XINUM.GT.100)     THEN
         WRITE   (*,8095)
         GO  TO  8045
      END  IF
 8095 FORMAT('TOO    MANY   INTERVALS    (100-MAX!).     TRY   AGAIN.')
C
```

```
 8000   CONTINUE
C
C---INPUT   INFO.  FOR  DIST.  OF  PROJECT   COST  (TOTAL).
C
        WRITE(*,8081)
        WRITE(*,8072)
 8072   FORMAT(//5X,'LOWER        AND   UPPER   ENDPOINTS    FOR   TOTAL   COST   DIST.:')
        LL(22)=INT(CTOT-18*SQRT(CTVAR))
        RR(22)=INT(CTOT+18*SQRT(CTVAR))
        IF  (LL(22).LT.0)      LL(22)=0
        IF  (LL(22).EQ.RR(22))        THEN
          LL(22)=LL(22)-10
          RR(22)=RR(22)+10
        ENDIF
        WRITE(*,8091)      LL(22),RR(22)
        WRITE(*,8070)
        READ(*,8074)      DCHAR
        IF  (DCHAR.EQ.'N'.OR.DCHAR.EQ.'n')             THEN
          WRITE(*,8071)
          READ(*,*)     LL(22),RR(22)
        ENDIF
C
 8020   WRITE(*,8081)
        WRITE(*,*)     'HOW   MANY   INTERVALS    FOR   THE   PROJECT    COST   DIST.?'
        WRITE(*,*)     '   --recommended       value    is  no   more   than   20'
        READ(*,*)     NCELLS(22)
        XINUM=NCELLS(22)
        IWIDTH(22)=(RR(22)-LL(22))/XINUM
        IF  (XINUM.GT.100)       THEN
          WRITE    (*,8095)
          GO   TO   8020
        ENDIF
C
C---ECHO    THE   DISTRIBUTION      VALUES    TO  THE   GENERAL    OUTPUT    FILE
C
        WRITE(3,*)     'VALUES    OF  ENDPOINTS    FOR  EMPIRICAL    DIST.'S    ARE:'
        WRITE(3,8098)
        WRITE(3,*)     '              LEFT-END          RIGHT-END           INT.-WIDTH'
        DO  8015   K=1,NKEYEN
          WRITE(3,16)      KEYEVN(K)
          WRITE(3,*)     '   REAL.   TIME   DIST.'
          WRITE(3,8096)      LL(K),RR(K),IWIDTH(K)
          WRITE(3,*)     '   COST    FUNC.'
          WRITE(3,8096)      LL(K+10),RR(K+10),IWIDTH(K+10)
          WRITE(3,8098)
 8015   CONTINUE
        WRITE(3,*)     'TOTAL    PROJECT    COST'
        WRITE(3,8096)      LL(22),RR(22),IWIDTH(22)
   16   FORMAT(1X,'KEY       EVENT    ',I5)
 8096   FORMAT(10X,I10,5X,I10,3X,F10.2)
 8098   FORMAT(/)
C
        CALL    SAMPLESIZE(NMCS)
```

```
C
C----INITIALIZE      FLAGONE----
      FLAGONE=0
C
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C     SUBROUTINE      I N T R O
C
C     THIS  SUBROUTINE   WRITES   INTRO.  INFORMATION    ON  THE  PROGRAM   AN-COST
C  TO  THE  SCREEN
C
C-----------------------------------------------------------------
C
      SUBROUTINE    INTRO
C
      WRITE(*,10)
 10   FORMAT(25(/))
      WRITE(*,1)
      PAUSE
      WRITE(*,10)
 19   WRITE(*,20)
 20   FORMAT(10X,'1.       GENERAL    PROGRAM    INFORMATION',/,
     &10X,'2.     CONTINUE    WITH   MONTE    CARLO    SIMULATION',//,
     &5X,'ENTER     CHOICE',//)
      READ(*,*)     ICH
      IF  (ICH.LT.1.OR.ICH.GT.2)        THEN
        WRITE(*,*)     'OUT   OF  RANGE'
        GOTO   19
      ENDIF
      IF  (ICH.EQ.2)     RETURN
C
      WRITE(*,6)
      PAUSE
      WRITE(*,2)
      PAUSE
      WRITE(*,3)
      PAUSE
      WRITE(*,4)
      PAUSE
      WRITE(*,5)
      PAUSE
      WRITE(*,7)
      PAUSE
      WRITE(*,10)
C
  1   FORMAT(29X,'PROGRAM                   AN-COST',////,
     *29X,'           WRITTEN    BY',////,
     *28X,'   RUSSELL      S.     VOG)MANN',///,
     *5X,'ADVISOR:       SALAH   E.  ELMAGHRABY',///)
  2   FORMAT(//10X,'TO       RUN   THIS   PROGRAM,     YOU   MUST:',/,
     *10X,'                                      ',/,
```

```
     *10X,'        1. CREATE   A FILE   CONTAINING    SOME  GENERAL    INFORMATION',/,
     *10X,'           AND  ACTIVITY   DURATION/COST      INFORMATION.    (THIS   CAN',/,
     *10X,'           BE  DONE   BY  RUNNING     PROGRAM    ANC-IN.EXE)',/,
     *10X,'        2. PLACE   THE  MEANS   AND  STANDARD    DEVIATIONS    OF  THE',/,
     *10X,'           KEY  EVENTS   IN  A  DATA   FILE   NAMED   "TRNS.ANC".',/,
     *10X,'           (THIS   CAN   DONE   BY  RUNNING    PROGRAM    DOD1TRNS.EXE',/,
     *10X,'            WHICH   IS  AN  ALTERATION    OF  DODIN'S    PROGRAM)     ',/,
     *10X,'        3. RUN  AN-COST.EXE.',///)
    3 FORMAT(//15X,'PROGRAM        ANC-IN.EXE     HAS  THE  ADDITIONAL     FACILITY',/,
     *10X,'FOR     RANDOMLY    GENERATING     ACTIVITY     NETWORKS.',//,
     *15X,'THE     SAME   FORMAT    IN  THE   DATA   FILE   ON  DURATION/COST      IS',/,
     *10X,'USED    AS  IS  IN  DODIN'S     PROGRAMS,     THEREFORE     DOD1TRNS.EXE',/,
     *10X,'WILL    GIVE   THE   SAME   INFORMATION     ON  THE  DISTRIBUTIONS      OF',/,
     *10X,'KEY     EVENTS    AS  DOES   DODIN1.EXE.',///)
    4 FORMAT(///15X,'FOR        FURTHER    DETAILS    ON  PROGRAMS    ANC-IN.EXE,     ',/,
     *10X,'DOD1TRNS.EXE,        OR  AN-COST.EXE      REFER   TO:',//,
     *10X,'"PROJECT      BIDDING    UNDER    CHANCE    TIME   CONSTRAINTS"',//,
     *10X,'Master's      Thesis    by  RUSSELL    S.  VOGTMANN',///)
    5 FORMAT(///15X,'LIMITATIONS            ON  THIS   PROGRAM:',//,
     *10X,'1.      MAX  100   ARCS',/,
     *10X,'2.      MAX  50  NODES',/,
     *10X,'3.      30 DURATIONS     FOR  AN  ACTIVITY',/,
     *10X,'4.      DIRECTED     ACYCLIC     NETWORKS',///)
    6 FORMAT(///10X,'THIS         PROGRAM     IS  DESIGNED     TO  DETERMINE     THE  ',/,
     *5X,'DISTRIBUTIONS        OF  REALIZATION      TIME   AND   COST   OF  KEY   EVENTS,',/,
     *5X,'AND    THE   STREAMS    OF  CASH   FLOW   OF  DIRECTED     ACYCLIC     NETWORKS.',/,
     *5X,'   ',/,
     *5X,'IT    DOES   THIS   BY  MONTE    CARLO   SIMULATION     OF  THE   NETWORK',////)
    7 FORMAT(///10X,'ONCE         THIS   PROGRAM    IS  COMPLETE,     PROGRAM    CASH.EXE',/,
     *5X,'CAN    BE  USED   TO  VIEW   THE  DISTRIBUTION      FUNCTIONS,     THE  CASH',/,
     *5X,'FLOW    STREAMS,     THE  PERCENTILES     OF  THE  DISTRIBUTIONS,      AND  TO',/,
     *5X,'DO    CASH   FLOW   CALCULATIONS      FOR  THE  PROBLEM',///)
C
      RETURN
      END
C-----------------------------------------------------------------------
C
C       SUBROUTINE       K E I N F O
C
C            THIS  SUBROUTINE    SENDS   THE   INFORMATION     ON  THE  DIST.   OF
C       THE  REALIZATION     TIME   OF  THE   KEY  EVENTS    TO  AN  OUTPUT    FILE
C       CALLED    KEDIST.DAT     .
C            THE   CRITICAL    INDICES    ARE   ALSO   SENT    TO  THE,   ABOVE,    FILE.
C
C-----------------------------------------------------------------
C
      SUBROUTINE     KEINFO(DF,MEAN,VAR,NKEYEN,LL,RR,IWIDTH,CI,NACT,
     *                      KEYEVN)
      INTEGER    NKEYEN,LL(22),RR(22),NACT,KEYEVN(10)
      REAL       DF(22,102),IWIDTH(22),CI(100)
      DOUBLE    PRECISION      MEAN(22),VAR(22)
C
      REAL      F
```

```
C
      WRITE(4,3088)
      DO 3009 J1=1,NKEYEN
C
      WRITE(4,*)
      WRITE(4,3083) KEYEVN(J1)
 3083 FORMAT(/1X,'     ---- KEY EVENT',I3,'     ------')
      KHOLD=(RR(J1)-LL(J1))/IWIDTH(J1)+2
      KHOLD=INT(KHOLD)
C
C---THE FOLLOWING <DO> LOOP SENDS THE EMPIRICAL DATA ON THE DENSITY
C---FUNCTION OF THE REALIZATION TIME OF THE KEY EVENTS TO FILE KEDIST.DAT
C
      DO 3010 J2=1,KHOLD
        IF (J2.EQ.1) THEN
          WRITE(4,3090) LL(J1),DF(J1,1)
          GO TO 3010
        ENDIF
        IF (J2.EQ.KHOLD) THEN
          WRITE(4,3092) RR(J1),DF(J1,KHOLD)
          GO TO 3010
        ENDIF
      X1=LL(J1)+(J2-2)*IWIDTH(J1)
      X2=X1+IWIDTH(J1)
      X3=DF(J1,J2)
      WRITE(4,3091) X1,X2,X3
C
 3010 CONTINUE
C
C---THE FOLLOWING <DO> LOOP SENDS THE DATA ON THE DISTRIBUTION FUNC.
C---TO THE FILE.
C
      SUM=0.00
      DO 3012 J2=1,KHOLD-1
       SUM=SUM+DF(J1,J2)
       X1=LL(J1)+(J2-1)*IWIDTH(J1)
       WRITE(4,3094) X1,SUM
 3012 CONTINUE
C
C---THE OBSERVED MEAN AND VARIANCE OF THE REALIZATION TIME OF THE
C---KEY EVENTS (OBTAINED FROM THE MONTE CARLO SIM.) IS ALSO SENT
C---TO THE FILE.
C
      WRITE(4,3095) MEAN(J1)
      WRITE(4,3096) VAR(J1)
      IF (VAR(J1).LE.0.) VAR(J1)=0.
      WRITE(4,3099) SQRT(VAR(J1))
C
 3009 CONTINUE
C
C---NOW SEND THE DIST. INFO. ON PROJECT COST TO THE FILE.
C
      WRITE(7,*)
```

```
      WRITE(7,*)
      WRITE(7,*)    '   ----  PROJECT   COST  ----    '
      KHOLD=(RR(22)-LL(22))/IWIDTH(22)+2
      KHOLD=INT(KHOLD)
C
C---THIS  <DO>  LOOP  SENDS  THE  DATA  ON  THE  DENSITY  FUNC.  OF  PROJECT
C    COST  TO  THE  FILE.
C
      DO  3050  J2=1,KHOLD
        IF  (J2.EQ.1)  THEN
          WRITE(7,3090)    LL(22),DF(22,1)
          GO  TO  3050
        ENDIF
        IF  (J2.EQ.KHOLD)  THEN
          WRITE(7,3092)    RR(22),DF(22,KHOLD)
          GO  TO  3050
        ENDIF
        X1=LL(22)+(J2-2)*IWIDTH(22)
        X2=X1+IWIDTH(22)
        X3=DF(22,J2)
        WRITE(7,3091)    X1,X2,X3
 3050 CONTINUE
C
C---THIS  LOOP  SENDS  THE  DATA  ON  THE  DIST.  FUNC.  TO  THE  FILE.
C
      SUM=0.00
      DO  3055  J2=1,KHOLD-1
        SUM=SUM+DF(22,J2)
        X1=LL(22)+(J2-1)*IWIDTH(22)
        WRITE(7,3094)    X1,SUM
 3055 CONTINUE
C
C---NOW  SEND  THE  OBSERVED  MEAN  AND  VAR.  OF  THE  PROJECT  COST.
C
      WRITE(7,3095)    MEAN(22)
      WRITE(7,3096)    VAR(22)
      IF  (VAR(22).LE.0.)   VAR(22)=0.
      WRITE(7,3099)    SQRT(VAR(22))
C
C---SENDING  COST  DIST.  INFORMATION  TO  OUTPUT  FILE  COSTD.DAT
C
      DO  3020  J1=1,NKEYEN
C
      WRITE(7,*)
      WRITE(7,3083)    KEYEVN(J1)
      KHOLD=(RR(J1+10)-LL(J1+10))/IWIDTH(J1+10)+2
      KHOLD=INT(KHOLD)
C
C---THE  FOLLOWING  <DO>  LOOP  SENDS  THE  EMPIRICAL  DATA  ON  THE  DENSITY
C---FUNCTION  OF  THE  COST  OF  THE  KEY  EVENTS  TO  FILE  COSTD.DAT
C
      DO  3021  J2=1,KHOLD
        IF  (J2.EQ.1)  THEN
```

```
            WRITE(7,3090)      LL(J1+10),DF(J1+10,1)
            GO  TO  3021
          ENDIF
          IF  (J2.EQ.KHOLD)      THEN
            WRITE(7,3092)      RR(J1+10),DF(J1+10,KHOLD)
            GO  TO  3021
          ENDIF
        X1=LL(J1+10)+(J2-2)*IWIDTH(J1+10)
        X2=X1+IWIDTH(J1+10)
        X3=DF(J1+10,J2)
        WRITE(7,3091)      X1,X2,X3
C
 3021   CONTINUE
C
C---THE    FOLLOWING    <DO>   LOOP   SENDS   THE   DATA   ON   THE   DISTRIBUTION     FUNC.
C---TO   THE   FILE.
C
        SUM=0.00
        DO  3022   J2=1,KHOLD-1
         SUM=SUM+DF(J1+10,J2)
         X1=LL(J1+10)+(J2-1)*IWIDTH(J1+10)
         WRITE(7,3094)      X1,SUM
 3022   CONTINUE
C
C---THE    OBSERVED    MEAN   AND   VARIANCE    OF   THE   REALIZATION     TIME   OF   THE
C---KEY    EVENTS    (OBTAINED     FROM   THE   MONTE    CARLO    SIM.)    IS   ALSO   SENT
C---TO    THE   FILE.
C
        WRITE(7,3095)      MEAN(J1+10)
        WRITE(7,3096)      VAR(J1+10)
        IF   (VAR(J1+10).LE.0.)        VAR(J1+10)=0.
        WRITE(7,3099)      SQRT(VAR(J1+10))
C
 3020   CONTINUE
C
C---THE    FOLLOWING    SENDS   THE   CRITICAL    INDICES    OF   THE   ACTIVITIES     TO
C---THE,    ABOVE,    DATA    FILE.
C
        WRITE(4,3098)
 3098   FORMAT(/1X,'      ----   CRITICAL     INDEXES     ----    ')
C
        DO  3029   J2=1,NACT
          WRITE(4,3097)      J2,CI(J2)
 3097     FORMAT(1X,'ACTIVITY         ',I2,1X,'HAS      A  CRITICAL     INDEX    =  ',F7.4)
 3029   CONTINUE
C
 3088   FORMAT(/1X,'DISTRIBUTION          INFO.   ABOUT    THE    KEY   EVENTS.')
 3090   FORMAT(1X,'Pr      (  X   <=',I5,'     )  =  ',F8.4)
 3092   FORMAT(1X,'Pr      (  X   >  ',I5,'     )  =  ',F8.4)
 3091   FORMAT(1X,'Pr      (',F9.3,'<      X  <=',F9.3,')      =',F8.4)
 3094   FORMAT(1X,'Pr      (  X  <=   ',F9.3,'     )  =  ',F8.4)
 3095   FORMAT(/1X,'THE       MEAN   OF  THE   DIST.    =',F15.4)
 3096   FORMAT(/1X,'THE       VARIANCE     OF  THE   DIST.    =',F15.4)
```

```
3099  FORMAT(/1X,'THE      STD.  DEV.  OF  THE  DIST.   =',F15.4)
C
      RETURN
      END
C---------------------------------------------------------------
C
C     SUBROUTINE     O U T H E L P
C
C     THIS  SUBROUTINE   PROVIDES   INFORMATION    ON  THE  OUTPUT   OF  AN-COST
C
C---------------------------------------------------------------
C
      SUBROUTINE    OUTHELP
C
      WRITE(*,1)
    1 FORMAT(25(/))
      WRITE(*,2)
    2 FORMAT(5X,'THE      FOLLOWING    OUTPUT    IS  AVAILABLE    ON  THE   RUN',//,
     *10X,'           FILE                         DATA',/,
     *10X,'1.     KEDIST.DAT       ---      i) EMPIRICAL    DENSITY    AND   DIST.',/,
     *10X,'                                    FUNCTIONS    ON  REALIZATION',/,
     *10X,'                                    TIME   OF   KEY   EVENTS',/,
     *10X,'                                ii) ESTIMATED    CRITICAL    INDICES',/,
     *10X,'2.     COSTD.DAT        ---      i) EMPIRICAL    DENSITY    AND   DIST.',/,
     *10X,'                                    FUNCTIONS    ON  COST   OF  KEY',/,
     *10X,'                                    EVENTS',/,
     *10X,'                                ii) EMPIRICAL    DENSITY    AND   DIST.',/,
     *10X,'                                    FUNCTIONS    ON  TOTAL   PROJECT    COST',/,
     *10X,'3.     <USER   NAMED>   ---  GENERAL    INFO.   ON  RUN',/,
     *10X,'4.     CASHFLOW.DAT     ---  DATA   TO  BE  USED   BY  PROGRAM    "CASH"',/)
      PAUSE
C
      WRITE(*,3)
    3 FORMAT(5(/),5X,'TO       VIEW   THE  DISTRIBUTION     DATA,   AS  WELL   AS  ',/,
     *5X,'EXAMINE     THE   CASH   FLOW   FOR   THE  PROJECT,    RUN   PROGRAM    "CASH".',/,
     *5X,'PROGRAM     CASH   WILL    ALSO   DO  CASH   FLOW  CALCULATIONS     FOR  THE',/,
     *5X,'PROJECT.',///)
      PAUSE
      WRITE(*,1)
C
      RETURN
      END
C---------------------------------------------------------------
C
C      SUBROUTINE     P A Y M E N T S
C
C          THIS  SUBROUTINE    DETERMINES    THE  DISBURSEMENTS     FOR  EACH
C      PERIOD   AND  EACH   MONTE   CARLO   SAMPLE.    AT  THE  END  OF  THE  MC
C      SIMULATION,    WE  WILL   HAVE   AN  AVERAGE    DISBURSEMENT     TO  USE   IN
C      THE  DETERMINATION     OF  PROFIT    MARGINS.
C
C---------------------------------------------------------------
C
```

```
      SUBROUTINE    PAYMENTS(KEYDUR,NACT,CYCLE,ACOST,DURATN,ES,EF,SDISBUR,
     *                   NMCS,MXPERD,CP1,GAMMA,SUBCO,SUBM,NKEYEN,PCOST)
      INTEGER   NACT,CYCLE,NMCS,MXPERD,NKEYEN
      REAL      SDISBUR(100),TDURAT,ACOST(100),DURATN(100),ES(100),
     *          EF(100),CP1(100),GAMMA,SUBM(100,10),KEYDUR(10)
      DOUBLE  PRECISION   SUBCO(11),PCOST
      INTEGER   CCPER
      REAL      DISBUR(100),SUBPAY(10,100)
C
      NPERD=KEYDUR(NKEYEN)/CYCLE+3
       IF(NPERD.GT.MXPERD)       MXPERD=NPERD
C
      DO  6001   I=1,NKEYEN
      DO  6002   J=1,NPERD
        SUBPAY(I,J)=0
 6002 CONTINUE
 6001 CONTINUE
C
      DO  6003   I=1,NKEYEN+1
        SUBCO(I)=0
 6003 CONTINUE
C
      DO  6004   I=1,100
        DISBUR(I)=0.0
 6004 CONTINUE
C
      DO  6010   I=1,NACT
        IF  (DURATN(I).EQ.0)      GO  TO  6010
        HSTART=ES(I)
        HFINSH=EF(I)
 6150   MP=HSTART/CYCLE+1
        MR=(MP-1)*CYCLE
        MS=MP*CYCLE
        IF  (MS.GT.HFINSH)      GO  TO  6160
C
          DISBUR(MP)=DISBUR(MP)+(MS-HSTART)*ACOST(I)/DURATN(I)
        DO  6005   I2=1,NKEYEN
          IF  (SUBM(I,I2).NE.0)       THEN
            SUBPAY(I2,MP)=SUBPAY(I2,MP)+(MS-HSTART)*ACOST(I)*
     *                          SUBM(I,I2)/DURATN(I)
          END  IF
 6005   CONTINUE
C
        HSTART=MP*CYCLE
        GO  TO  6150
C
 6160   IF  (MR.LT.HFINSH)      THEN
C
          IF  (MR.LT.HSTART)      THEN
            DISBUR(MP)=DISBUR(MP)+(HFINSH-HSTART)*ACOST(I)/DURATN(I)
            DO  6165   I2=1,NKEYEN
              IF  (SUBM(I,I2).NE.0)       THEN
                SUBPAY(I2,MP)=SUBPAY(I2,MP)+(HFINSH-HSTART)*ACOST(I)*
```

```
     *                              SUBM(I,I2)/DURATN(I)
                 END  IF
 6165         CONTINUE
C
            ELSE
              DISBUR(MP)=DISBUR(MP)+(HFINSH-MR)*ACOST(I)/DURATN(I)
              DO  6167   I2=1,NKEYEN
                IF  (SUBM(I,I2).NE.0)        THEN
                  SUBPAY(I2,MP)=SUBPAY(I2,MP)+(HFINSH-MR)*ACOST(I)*
     *                              SUBM(I,I2)/DURATN(I)
                 END  IF
 6167         CONTINUE
            END  IF
C
          END  IF
 6010   CONTINUE
C
C---THE   FOLLOWING   <DO>   LOOP   ADDS   IN  THE   COST   CP1(   ),  WHICH   IS  THE
C---CONSTANT     COST   THAT   THE   ACTIVITY    INCURS    REGARDLESS    OF  DURATION
C
        DO  6020   J=1,NACT
          CCPER=INT(ES(J)/CYCLE)+1
C
          DISBUR(CCPER)=DISBUR(CCPER)+CP1(J)
          DO  6125   J2=1,NKEYEN
            IF  (SUBM(J,J2).NE.0)        THEN
              SUBPAY(J2,CCPER)=SUBPAY(J2,CCPER)+CP1(J)*SUBM(J,J2)
            END  IF
 6125     CONTINUE
C
 6020   CONTINUE
C
C  ---   CHECK   ALGORITHM    FOR   FUTURE    VALUE    HERE!!!!
C
        DO  6040   I=1,NPERD
          DISBUR(I)=DISBUR(I)*(1+GAMMA*CYCLE*(I-1)/365)
 6040   CONTINUE
C
        DO  6042   I=1,NKEYEN
        DO  6043   J=1,NPERD
          SUBPAY(I,J)=SUBPAY(I,J)*(1+GAMMA*CYCLE*(J-1)/365)
 6043   CONTINUE
 6042   CONTINUE
C
        DO  6028   I=1,NKEYEN
        DO  6029   J=1,NPERD
          SUBCO(I)=SUBCO(I)+SUBPAY(I,J)
 6029   CONTINUE
 6028   CONTINUE
C
C---DETERMINE      TOTAL    PROJECT    COST
C
        PCOST=0
```

```
      DO  3500   I=1,NKEYEN
        PCOST=PCOST+SUBCO(I)
 3500 CONTINUE
C
      DO  6030   J=1,NPERD
        SDISBUR(J)=SDISBUR(J)+DISBUR(J)/NMCS
 6030 CONTINUE
C
      RETURN
      END
C-----------------------------------------------------------------
C
C       SUBROUTINE      P C I N F O
C
C       THIS  SUBROUTINE   GIVES   INFORMATION    ON  THE  ALLOCATION    OF  COST  OF
C       COMMON   ACTIVITIES.
C
C-----------------------------------------------------------------
C
      SUBROUTINE    PCINFO
C
      WRITE(*,1)
    1 FORMAT(25(/))
      WRITE(*,2)
    2 FORMAT(5X,'THE      COST   OF  COMMON   ACTIVITIES    CAN  BE  ALLOCATED    TO  ',/,
     *5X,'ITS     KEY  EVENTS   BY  THE  FOLLOWING    METHODS:',//,
     *15X,'1.     PROBABILITY     METHOD',/,
     *15X,'2.     USER   GIVEN    %'S',///,
     *1X,'INPUT     INFORMATION     OPTION...(OTHER      VALUES---RETURN)',///)
C
      READ(*,*)    IOPTION
      IF  (IOPTION.LT.1.OR.IOPTION.GT.2)          RETURN
C
      IF  (IOPTION.EQ.1)      THEN
        WRITE(*,1)
        WRITE(*,3)
    3   FORMAT(5X,'THIS      COST   ALLOCATION    METHOD    DETERMINES     THE',/,
     *5X,'PERCENTAGES      OF  THE   KEY   EVENTS    AS  FOLLOWS...',///,
     *10X,'WEIGHT(i)      =  Pr(  X(i)<=X(j1)     )  *...*   Pr(  X(i)<=X(jN)     )  ',//,
     *10X,'   --where   X(i)  denotes    the   r.v.   of  key   event    i',//,
     *5X,'PERCENTAGE(i)       =  NORMALIZED      VALUE',//)
        PAUSE
        WRITE(*,4)
    4   FORMAT(///5X,'FOR       MORE   INFORMATION     SEE   MASTER'S     THESIS',//,
     *5X,'"PROJECT      BIDDING    UNDER    CHANCE    TIME    ESTIMATES"',/,
     *5X,'by     Russell    S.  Vogtmann.',//)
        PAUSE
C
      ELSE
        WRITE(*,1)
        WRITE(*,5)
    5   FORMAT(5X,'THIS      OPTION    ALLOWS    THE   USER   TO  INPUT    THE',/,
     *5X,'PERCENTAGES       USING   HIS   OWN   DISCRETION.',///)
```

```
          PAUSE
        ENDIF
C
        RETURN
        END
C-----------------------------------------------------------------
C
C       SUBROUTINE      R N G E N E R A T O R
C
C           THIS  SUBROUTINE  GENERATES  THE  RANDOM  NUMBERS  NEEDED
C       FOR  THE  MONTE  CARLO  SIMULATION.
C
C-----------------------------------------------------------------
C
        SUBROUTINE    RNGENERATOR    (NACT,DURATN,DNUM,P1,P2,P3,P4,ACOST,CDNUM,
     *                                CP1,CP2,CP3,CP4,CP5,DIST,DURA,NOP,IX)
C
          INTEGER    NACT,DNUM(100),CDNUM(100),NOP(100)
          REAL    DURATN(100),P1(100),P2(100),P3(100),P4(100),CP1(100),
     *            CP2(100),CP3(100),CP4(100),ACOST(100),DIST(100,30),
     *            DURA(100,30),CP5(100)
          REAL*8    IX
C
C---THIS   LOOP   OBTAINS   VALUES   FOR   DURATN   (THE   DURATIONS   OF  ALL  THE
C---ACTIVITIES     IN  THE   NETWORK).    IT  ALSO  CALLS  SUBROUTINE   RCOST   TO
C---OBTAIN    THE   RANDOM   VALUES   OF  THE   COST   OF  THE   ACTIVITIES.
C
        DO  9000   I=1,NACT
C
          GOTO   (9011,9012,9013,9014,9015,9016,9017),DNUM(I)
C
9011              CALL   UNICON(P3(I),P4(I),DURATN(I),IX)
                  GO  TO  8999
9012              CALL   TRIANGULAR(P3(I),P1(I),P4(I),DURATN(I),IX)
                  GO  TO  8999
9013              CALL   RNORMAL(P1(I),P2(I),P3(I),P4(I),DURATN(I),IX)
                  GO  TO  8999
9014              CALL   EXPON(P1(I),P2(I),P3(I),P4(I),DURATN(I),IX)
                  GO  TO  8999
9015              CALL   GAMMA(P1(I),P2(I),P3(I),P4(I),DURATN(I),IX)
                  GO  TO  8999
9016              CALL   BETA(P1(I),P2(I),P3(I),P4(I),DURATN(I),IX)
                  GO  TO  8999
9017              CALL   USER1(NOP,DIST,DURA,I,DURATN(I),IX)
C
C---HERE,    I  CALL   SUBROUTINE   RCOST   TO  OBTAIN   THE   RANDOM   VALUES   OF
C---THE    COST   OF  THE   ACTIVITIES.
C
 8999   CALL   RCOST(DURATN(I),ACOST(I),CDNUM(I),CP2(I),CP3(I),CP4(I),
     *               CP5(I),IX)
C
 9000   CONTINUE
C
```

```
      RETURN
      END
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C     S U B R O U T I N E S
C
C        THE FOLLOWING PROGRAM AREA CONTAINS THE VARIOUS SUBROUTINES
C     TO GENERATE RANDOM NUMBERS FROM MANY DIFFERENT DISTRIBUTIONS.
C
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C     NORMAL DIST.         MEAN=XBAR        VARIANCE=VAR
C                          NMIN=MIN. VALUE  NMAX=MAX. VALUE
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      SUBROUTINE RNORMAL(XBAR,VAR,NMIN,NMAX,VALUE,IX)
      REAL   XBAR,VAR,NMIN,NMAX,VALUE
      REAL*8   IX
C
C---POLAR COORDINATE METHOD
C
9020  CALL RANDOM(IX,R1)
      CALL RANDOM(IX,R2)
      VALUE=XBAR+SQRT(-2*ALOG(R1)*VAR)*COS(2*3.14159*R2)
      IF (VALUE.LT.NMIN.OR.VALUE.GT.NMAX)        GOTO 9020
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C       UNIFORM--CONTINUOUS          A=LEFT END  B=RIGHT END
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      SUBROUTINE UNICON(A,B,VALUE,IX)
      REAL     A,B,VALUE
      REAL*8   IX
C
      CALL RANDOM(IX,X)
      VALUE=A+(B-A)*X
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C       TRIANGULAR  DIST.       A=LEFT END  B=MODE  C=RIGHT END
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      SUBROUTINE TRIANGULAR(A,B,C,VALUE,IX)
      REAL     A,B,C,VALUE
      REAL*8   IX
C
```

```
      CALL   RANDOM(IX,X)
      IF(X.LE.(B-A)/(C-A))        THEN
        VALUE=A+SQRT((C-A)*(B-A)*X)
      ELSE
        VALUE=C-SQRT((C-A)*(C-B)*(1-X))
      ENDIF
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C     GAMMA   DIST.    ---     GAMMA(ALPHA,BBETA)
C
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      SUBROUTINE     GAMMA(ALPHA,BBETA,NMIN,NMAX,DURATN,IX)
      REAL       BBETA,ALPHA,NMIN,NMAX,DURATN
      REAL*8     IX
C
      X=1.0
      K=INT(ALPHA)
      Q=ALPHA-K
      IF  (K.LE.0)    THEN
        CALL   RANDOM(IX,U)
        Z=-ALOG(U)
        CALL   BETA(Q,1-Q,0,1,Y,IX)
        DURATN=BBETA*(X+Y*Z)
        RETURN
      ELSE
        CALL   RANDOM(IX,U)
        TEMP=U
        IF  (K.GT.1)    THEN
          DO  9025   I=1,K-1
            CALL   RANDOM(IX,U)
            TEMP=TEMP*U
 9025     CONTINUE
        ENDIF
        X=-ALOG(TEMP)
        IF  (Q.EQ.0)    THEN
          DURATN=BBETA*X
          RETURN
        ELSE
          CALL   RANDOM(IX,U)
          Z=-ALOG(U)
          CALL   BETA(Q,1-Q,0,1,Y,IX)
          DURATN=BBETA*(X+Y*Z)
        ENDIF
      ENDIF
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C     EXPONENTIAL     --      EX(DMEAN)
C
```

```
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      SUBROUTINE     EXPON(DMEAN,DUMMY,NMIN,NMAX,DURATN,IX)
      REAL         DMEAN,DUMMY,NMIN,NMAX,DURATN
      REAL*8       IX
C
 9055 CALL   RANDOM(IX,X)
C
      DURATN=-ALOG(X)*DMEAN
      IF(DURATN.LT.NMIN.OR.DURATN.GT.NMAX)              GOTO   9055
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C     BETA   DIST.         C=LEFT    ENDPOINT        D=RIGHT    ENDPOINT
C                          A=1ST    SHAPE    PAR.     B=2ND    SHAPE    PAR.
C
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      SUBROUTINE     BETA(A,B,C,D,DURATN,IX)
      REAL       A,B,C,D,DURATN
      REAL*8       IX
      ITER=1
C
 9057 CALL   RANDOM(IX,U1)
      CALL   RANDOM(IX,U2)
      Y1=U1**(1/A)
      Y2=U2**(1/B)
      IF  ((Y1+Y2).LE.1.0)       THEN
         Y=Y1/(Y1+Y2)
         DURATN=C+(D-C)*Y
         RETURN
      ELSE
         ITER=ITER+1
      ENDIF
C
      IF  (ITER.LT.100)       THEN
        GOTO   9057
      ELSE
        WRITE(*,9058)
        STOP
      ENDIF
 9058 FORMAT(1X,     ' 100   INTERATIONS     OVER    IN   BETA    WITHOUT    SUCCESS')
      RETURN
      END
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C     USER   DEFINED    DISCRETE    DENSITIES
C
C           THESE    ARE   ORDERED    PAIRS    OF  DURATIONS     AND   PROBABILITIES--
C     THE    DURATIONS     ARE   STORED    IN   DURA,    AND    THE    PROB.'S    IN  DIST
C
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
C
      SUBROUTINE     USER1(NOP,DIST,DURA,I,DURATN,IX)
      INTEGER    I,NOP(100)
      REAL       DIST(100,30),DURA(100,30),DURATN
      REAL*8     IX
C
      CALL   RANDOM(IX,X)
      CUMDIST=0.0
      DO  9060   II=1,NOP(I)
        CUMDIST=CUMDIST+DIST(I,II)
        IF  (X.LE.CUMDIST)       THEN
          DURATN=DURA(I,II)
          RETURN
        ENDIF
 9060 CONTINUE
 9065 RETURN
      END
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C         RANDOM    UNIFORM(0,1)
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      SUBROUTINE     RANDOM(IX,D)
      DOUBLE    PRECISION     Y,A,P,IX,B15,B16,XHI,XALO,LEFTLO,FHI,K
      DATA   A/16807.D0/,B15/32768.D0/,B16/65536.D0/,P/2147483647.D0/
C
      XHI=IX/B16
      XHI=XHI-DMOD(XHI,1.D0)
      XALO=(IX-XHI*B16)*A
      LEFTLO=XALO/B16
      LEFTLO=LEFTLO-DMOD(LEFTLO,1.D0)
      FHI=XHI*A+LEFTLO
      K=FHI/B15
      K=K-DMOD(K,1.D0)
      IX=(((XALO-LEFTLO*B16)-P)+(FHI-K*B15)*B16)+K
      IF(IX.LT.0.D0)       IX=IX+P
      Y=IX*4.656612875E-10
      D=REAL(Y)
      IX=IX+1.0
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C        SUBROUTINE       R  C  O  S  T
C
C           THIS   SUBROUTINE     DETERMINES     THE   COST   OF  AN  ACTIVITY
C     IN  A  PARTICULAR     MONTE    CARLO    SAMPLE.    (TAKING    INTO   ACCOUNT
C     THE   POSSIBLE     RANDOM    VARIATION     OF  COST).
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
```

```
      SUBROUTINE   RCOST(DURATN,ACOST,CDNUM,CP2,CP3,CP4,CP5,IX)
      INTEGER   CDNUM
      REAL      DURATN,ACOST,CP2,CP3,CP4,CP5
      REAL*8    IX
C
      GOTO(9071,9072,9073,9074),CDNUM
C
C---CONSTANT   COST/UNIT   TIME
 9071 ACOST=CP2*DURATN
      RETURN
C
C---COST=NORMAL   R.V.  ABOUT  A MEAN=CONST.*DURATION   AND  A
C---SPECIFIED   VARIANCE.
 9072 CALL  RNORMAL(CP2*DURATN,CP3,CP4,CP5,ACOST,IX)
      RETURN
C
C---COST=UNIFORM   R.V.  ABOUT  A  MEAN=CONST.*DURATION   AND  SPECIFIED
C---DISTANCE   BETWEEN  ENDPOINTS   AND  MEAN.
 9073 CALL  UNICON(CP2*DURATN-CP3,CP2*DURATN+CP4,ACOST,IX)
      RETURN
C
C---TRIANGULAR   COST  DIST.  ABOUT  A  MODE=CONST.*DURATION   AND  SPECIFIED
C---DISTANCE   BETWEEN  ENDPOINTS   AND  MODE
 9074 CALL  TRIANGULAR(CP2*DURATN,CP2*DURATN-CP4,CP2*DURATN+CP5,
     *                     ACOST,IX)
      RETURN
      END
C-----------------------------------------------------------------
C
C     SUBROUTINE   S A M P H E L P
C
C     THIS  SUBROUTINE   PROVIDES   INFORMATION   ON  THE  SAMPLE  SIZE  DETERMINA-
C     TION.
C
C-----------------------------------------------------------------
C
      SUBROUTINE   SAMPHELP
C
      WRITE(*,1)
    1 FORMAT(25(/))
      WRITE(*,2)
    2 FORMAT(5X,'THIS    PROGRAM   IS  ESTIMATING   THE  DISTRIBUTION    FUNC-',/,
     *5X,'TIONS   OF  DURATION   AND  COST  AT  VARIOUS   KEY  EVENTS.   IN  ',/,
     *5X,'ORDER   TO  DO  THIS  WITH  SUFFICIENT   ACCURACY,   WE  MUST  MAKE',/,
     *5X,'THE  SAMPLE  SIZE  LARGE  ENOUGH.',//,
     *5X,'THIS   SAMPLE  SIZE  CAN  BE  DETERMINED   VIA  THE  KOLMOGOROV-',/,
     *5X,'SMIRNOV   STATISTIC   FOR  GOODNESS-OF-FIT.     THIS  PROGRAM',/,
     *5X,'USES  THE  LIMITING  STATISTICS   (AS  SAMPLE  SIZE  INCREASES',/,
     *5X,'TO  INFINITY)  TO  DETERMINE   THE  SAMPLE  SIZE.',//,
     *5X,'THESE   VALUES  WERE  TABULATED   BY  N.  SMIRNOV   IN  "TABLE  FOR',/,
     *5X,'ESTIMATING   THE  GOODNESS   OF  FIT  OF  EMPIRICAL   DISTRIBUTIONS"',/,
     *5X,'ANNALS   OF  MATHEMATICAL   STATISTICS,   19(1948),   p.279-281',//)
      PAUSE
```

```
      WRITE(*,1)
C
      RETURN
      END
C--------------------------------------------------------------
C
C     SUBROUTINE     S A M P L E S I Z E
C
C     THIS    SUBROUTINE     DETERMINES    THE   REQUIRED    SAMPLE    SIZE    FOR    THE
C     REQUIREMENTS     OF   THE   USER.
C
C--------------------------------------------------------------
C
      SUBROUTINE     SAMPLESIZE(NMCS)
C
      INTEGER    NMCS
      REAL   D(11),A(11)
      CHARACTER*1      NCHAR
      DATA   D  /  0.826,0.861,0.895,0.932,0.973,1.02,1.073,1.138,1.225,
     *1.36,1.63     /
      DATA   A  /  0.5,0.55,0.6,0.65,0.7,0.75,0.8,0.85,0.9,0.95,0.99                /
C
 8099 WRITE(*,8100)
 8100 FORMAT(25(/))
C
      WRITE(*,8105)
 8105 FORMAT(5X,'THIS       PORTION     OF   THE    PROGRAM     DETERMINES      THE    NECESSARY',/
     *,5X,'SAMPLE      SIZE    FOR    THE    MONTE     CARLO    SIMULATION',///,
     *10X,'1.      SAMPLE    SIZE   INFORMATION',/,
     *10X,'2.      PROCEED    WITH    SAMPLE    SIZE    DETERMINATION',/,
     *10X,'3.      TERMINATE      PROGRAM',//,
     *5X,'ENTER     CHOICE',///)
      READ(*,*)     ICH
      IF   (ICH.LT.1.OR.ICH.GT.3)          THEN
        WRITE(*,*)      'OUT    OF    RANGE     --   TRY    AGAIN    CLOWN'
        GOTO    8099
      ENDIF
C
      IF   (ICH.EQ.3)      STOP
      IF   (ICH.EQ.1)      THEN
        CALL    SAMPHELP
        GOTO    8099
      ENDIF
C
      WRITE(*,8100)
C
      WRITE(*,8106)
 8106 FORMAT(///,5X,'INPUT        THE    MAXIMUM     DIFFERENCE      BETWEEN     THE    TRUE',/,
     *5X,'PROBABILITY        DISTRIBUTIONS        AND   THE    SAMPLED     DISTRIBUTIONS',/,
     *5X,'(e.g.      .01)',///)
      READ(*,*)      DK
 8109 WRITE(*,8110)
 8110 FORMAT(///,5X,'INPUT        THE    ALPHA     LEVEL     (FOR    A   1-ALPHA     CONFIDENCE)',/
```

```
     *,5X,'--ALLOWABLE      RANGE   IS  FROM   .01   TO  .5)',//)
      READ(*,*)     RAL
      IF  (RAL.LT.0.01.OR.RAL.GT.0.5)           THEN
        WRITE(*,*)     'OUT   OF  RANGE'
        WRITE(*,8100)
        GOTO   8109
      ENDIF
C
      VAL=0.0
      IF  (RAL.LE.0.5.AND.RAL.GT.0.45)           THEN
        CALL   SSZ(D(1),D(2),A(1),A(2),RAL,VAL)
      ELSE   IF  (RAL.LE.0.45.AND.RAL.GT.0.40)           THEN
        CALL   SSZ(D(2),D(3),A(2),A(3),RAL,VAL)
      ELSE   IF  (RAL.LE.0.40.AND.RAL.GT.0.35)           THEN
        CALL   SSZ(D(3),D(4),A(3),A(4),RAL,VAL)
      ELSE   IF  (RAL.LE.0.35.AND.RAL.GT.0.30)           THEN
        CALL   SSZ(D(4),D(5),A(4),A(5),RAL,VAL)
      ELSE   IF  (RAL.LE.0.30.AND.RAL.GT.0.25)           THEN
        CALL   SSZ(D(5),D(6),A(5),A(6),RAL,VAL)
      ELSE   IF  (RAL.LE.0.25.AND.RAL.GT.0.20)           THEN
        CALL   SSZ(D(6),D(7),A(6),A(7),RAL,VAL)
      ELSE   IF  (RAL.LE.0.20.AND.RAL.GT.0.15)           THEN
        CALL   SSZ(D(7),D(8),A(7),A(8),RAL,VAL)
      ELSE   IF  (RAL.LE.0.15.AND.RAL.GT.0.10)           THEN
        CALL   SSZ(D(8),D(9),A(8),A(9),RAL,VAL)
      ELSE   IF  (RAL.LE.0.10.AND.RAL.GT.0.05)           THEN
        CALL   SSZ(D(9),D(10),A(9),A(10),RAL,VAL)
      ELSE   IF  (RAL.LE.0.05.AND.RAL.GE.0.01)           THEN
        CALL   SSZ(D(10),D(11),A(10),A(11),RAL,VAL)
      ENDIF
C
      NMCS=(VAL/DK)**2
      WRITE(*,8100)
      WRITE(*,8115)     NMCS
 8115 FORMAT(5X,'THE     SAMPLE    SIZE    NECESSARY     TO  SATISFY    THE  GIVEN',/,
     &5X,'REQUIREMENTS        IS   ',I6,'.      DO  YOU  WANT    THE    MONTE    CARLO',/,
     &5X,'SIMULATION       TO  TAKE    THIS    MANY   SAMPLES     (Y  or  N)',//)
      READ(*,8120)     NCHAR
 8120 FORMAT(A1)
C
      IF  (NCHAR.EQ.'N'.OR.NCHAR.EQ.'n')          THEN
        WRITE(*,8125)
 8125   FORMAT(///,'THEN       INPUT    THE   NUMBER    YOU   WANT    TO  TAKE',/)
        READ(*,*)     NMCS
      ENDIF
C
      RETURN
      END
C-------------------------------------------------------------------
C
C        SUBROUTINE      S  E  P  A  R  A  T  E
C
C              THIS  SUBROUTINE    SEPARATES    THE   COST   OF  ACTIVITIES    CON-
```

```
C         TAINED    IN  THE   SUBGRAPHS    OF  MORE   THAN   ONE  KEY  EVENT   (I.E.   COM-
C         MON  ACTIVITIES).
C
C         ---THE   USER  CAN  CHOOSE   THE  PERCENTAGES    TO ATTRIBUTE    TO  EACH
C            KEY  ACTIVITY,    OR  HE  CAN  LET  THE  PROGRAM   SEPARATE   THE  COST
C            AS  FOLLOWS:
C
C              WEIGHT(i)=    Pr  (  X(i)<=X(j1)    ) *  ...  * Pr  (  X(i)<=X(jN)     )
C
C             **PERCENTAGE(I)=NORMALIZED          VALUES   OF  WEIGHT(i)
C
C        --THIS   METHOD   ASSUMES   THAT  THE  REALIZATION    TIME  PDF'S  ARE
C        --NORMALLY    DISTRIBUTED.     -->  SEE  MASTER'S   THESIS   FOR  DETAILS
C
C------------------------------------------------------------------
C
        SUBROUTINE    SEPARATE   (SUBM,J2,NKEYEN,NODEST,NODEFI,KEYEVN,AVERG,
     *                          STDV,PERCEN,NACT,ACOST)
        INTEGER    J2,NKEYEN,NODEST(100),NODEFI(100),KEYEVN(10),NACT
        REAL       SUBM(100,10),AVERG(10),STDV(10),PERCEN(100,10),ACOST(100)
C
        INTEGER    POINTER(10),METHOD,FLAG,NUM
        REAL       W(10),WTOT,TEMP
C
C--INITIALIZE      ARRAY   PERCEN   TO  ZERO
C
        DO  2000   J3=1,NACT
        DO  2001   J1=1,NKEYEN
          PERCEN(J3,J1)=0.
C
C---DO   HOUSEKEEPING    ON  MATRIX   SUBM
C
          IF  (SUBM(J3,J1).LT.0.000001)          SUBM(J3,J1)=0.0
C
 2001   CONTINUE
 2000   CONTINUE
C
        DO  2002   J2=1,NACT
C
          DO  2025   II=1,NKEYEN
            POINTER(II)=0
            W(II)=0
 2025     CONTINUE
C
C---DETERMINE      IF  ACTIVITY   IS  COMMON   TO  MORE   THAN   ONE  KEY  EVENT
C
          TOTAL=0
          DO  2003   J4=1,NKEYEN
            TOTAL=TOTAL+SUBM(J2,J4)
 2003     CONTINUE
C
C-----GO    TO  NEXT   ACTIVITY    IF  UNIQUE   TO  ONE  KEY  EVENT
C
```

```
      IF (TOTAL.LE.ACOST(J2))      GOTO  2002
C
 2089 WRITE(*,2090)    J2,NODEST(J2),NODEFI(J2)
 2090 FORMAT(//1X,'    ACTIVITY   ',I2,'   FROM  NODE  ',I4,'   TO  NODE   ',I4)
      WRITE(*,2091)
      WRITE(*,2092)
 2091 FORMAT(1X,'      ')
 2092 FORMAT(1X,'    is  common  to  the  following    KEY  EVENTS:')
C
      DO  2005   J3=1,NKEYEN
        IF  (SUBM(J2,J3).NE.0.0)       THEN
          POINTER(J3)=1
          PERCEN(J2,J3)=1
          WRITE(*,2093)     KEYEVN(J3)
        ENDIF
 2005 CONTINUE
 2093 FORMAT(1X,I4)
C
 2140 WRITE(*,2091)
      WRITE(*,2094)
      WRITE(*,2091)
      WRITE(*,2095)
      WRITE(*,2096)
      WRITE(*,2098)
 2094 FORMAT(//1X,'WHAT       METHOD   OF  SEPARATION    DO  YOU  WANT?')
 2095 FORMAT(1X,'      PROBABILITY     METHOD    -----    1')
 2096 FORMAT(1X,'      USER  GIVEN   %'S    --------    2')
 2098 FORMAT(1X,'      **information**      -------    10')
C
      READ(*,*)     METHOD
C
      IF  (METHOD.EQ.10)      THEN
        CALL   PCINFO
        GOTO   2089
      ENDIF
C
      IF  (METHOD.LT.1.OR.METHOD.GT.2)          THEN
        WRITE(*,2091)
        WRITE(*,2144)
 2144   FORMAT(5X,'value       OUT-OF-RANGE       ---     try   again')
        GOTO   2140
      ENDIF
C
C-----------USER       DECIDING    HIS   OWN   SEPARATION     PERCENTAGES--------------
C
      IF  (METHOD.EQ.2)      THEN
 2050   DO  2010   J3=1,NKEYEN
          IF  (POINTER(J3).EQ.1)       THEN
            WRITE(*,2097)      KEYEVN(J3)
            READ(*,*)     PERCEN(J2,J3)
          ENDIF
 2010   CONTINUE
C
```

```
          SU=0
          DO  2011   J3=1,NKEYEN
            PERCEN(J2,J3)=PERCEN(J2,J3)/100
            SU=SU+PERCEN(J2,J3)
 2011     CONTINUE
C
          IF  (SU.NE.1)    THEN
            WRITE(*,2099)
            GO  TO  2050
          ENDIF
        ENDIF
C
 2099   FORMAT(1X,'DOES      NOT  ADD  TO  1 ---  TRY  AGAIN')
 2097   FORMAT(1X,'Input       percentage     for   KEY  EVENT',I3)
C
C-----------------------PROBABILITY              METHOD-------------------------
C
        IF  (METHOD.EQ.1)     THEN
C
C---DETERMINE      KEY  EVENT  THAT  HAS  SMALLEST   MEAN,  MARK  WITH  FLAG
C
          TEMP=100000000.00
          DO  2100  J3=1,NKEYEN
            IF  (POINTER(J3).NE.0)       THEN
              IF  (AVERG(J3).LT.TEMP)       THEN
                TEMP=AVERG(J3)
                FLAG=J3
              ENDIF
            ENDIF
 2100     CONTINUE
C
C---DETERMINE      Pr  {  X(s)   <=  X(i)   }
C
          DO  2105  J3=1,NKEYEN
            IF  (POINTER(J3).NE.0       .AND.  J3.NE.FLAG)     THEN
              Z=(AVERG(J3)-AVERG(FLAG))/(SQRT(STDV(FLAG)**2+STDV(J3)**2))
C
C---PHIZ    DENOTES   THE  Pr  {  X(s)   <=  X(i)   }
C
              CALL   STDNOR(Z,PHIZ)
C
C---PLACE    NO  WEIGHT   ON  KEY  EVENT   i  IF   Pr  {  X(s)   <=  X(i)   }  >=  0.9
C---(THIS    IS  AN  ARBITRARY    SETTING)
C
              IF  (PHIZ.GE.0.9)      THEN
                POINTER(J3)=0
                PERCEN(J2,J3)=0
              ENDIF
C
            ENDIF
 2105     CONTINUE
C
C---DETERMINE      WEIGHT(I)    AS  SHOWN   IN  PREAMBLE    TO  THIS   SUBROUTINE
```

```
C---(   SEE   ABOVE   )
C
      DO  2160   J5=1,NKEYEN
        IF  (POINTER(J5).NE.1)       GOTO   2160
        W(J5)=1
      DO  2165   J6=1,NKEYEN
        IF  (J6.EQ.J5.OR.POINTER(J6).NE.1)           GOTO   2165
C
          Z=(AVERG(J6)-AVERG(J5))/(SQRT(STDV(J6)**2+STDV(J5)**2))
          CALL   STDNOR(Z,PHIZ)
C
          W(J5)=W(J5)*PHIZ
C
 ?'65  CONTINUE
 2160  CONTINUE
C
        WTOT=0.0
        DO  2030   J3=1,NKEYEN
          WTOT=WTOT+W(J3)
 2030   CONTINUE
C
        DO  2115   J3=1,NKEYEN
          IF  (POINTER(J3).NE.0)        THEN
            W(J3)=W(J3)/WTOT
            PERCEN(J2,J3)=W(J3)
          ENDIF
 2115   CONTINUE
C
       WRITE(*,2189)      J2
       WRITE(3,2189)      J2
       WRITE(3,2191)
       WRITE(*,2191)
       DO  2120   J3=1,NKEYEN
         IF  (PERCEN(J2,J3).EQ.0)         GO  TO  2120
         WRITE(*,2190)      100*PERCEN(J2,J3),KEYEVN(J3)
         WRITE(3,2190)      100*PERCEN(J2,J3),KEYEVN(J3)
 2120  CONTINUE
 2191  FORMAT(1X,'PERCENTAGE         TO  KEY   EVENTS   OF  COMMON    ACT.')
 2190  FORMAT(F6.2,1X,'PERCENT          TO  KEY   EVENT--',I4)
 2189  FORMAT(1X,'ACTIVITY        ',I2)
      ENDIF
C
C---CONTINUE      ORIGINAL    DO  LOOP   OVER   ACTIVITIES
C
 2002  CONTINUE
C
      RETURN
      END
C-----------------------------------------------------------------
C
C         SUBROUTINE       S  M  C  O  R  R
C
C       THIS   SUBROUTINE    KEEPS   TRACK   OF  THE   SUM   AND   THE   SUM   OF   SQUARES    OF
```

```
C       THE  COST  AND  DURATION   OF THE  KEY  EVENTS   TO  BE  USED   LATER   TO  GET
C       VALUES   OF  SAMPLE   CORRELATION    AND  A  LEAST   SQUARES    FIT  OF  THE  DATA.
C
C------------------------------------------------------------------
C
        SUBROUTINE     SMCORR(PCOST,SUBCO,KEYDUR,NKEYEN,SSUM,SSUMSQ,SUMXY)
C
        INTEGER    NKEYEN
        REAL   KEYDUR(10)
        DOUBLE    PRECISION     SSUM(21),SSUMSQ(21),SUMXY(11),SUBCO(11),PCOST
C
        DO  3600   I=1,NKEYEN
           SSUM(I)=SSUM(I)+KEYDUR(I)
           SSUMSQ(I)=SSUMSQ(I)+KEYDUR(I)*KEYDUR(I)
           SUMXY(I)=SUMXY(I)+KEYDUR(I)*SUBCO(I)
           SSUM(I+10)=SSUM(I+10)+SUBCO(I)
           SSUMSQ(I+10)=SSUMSQ(I+10)+SUBCO(I)*SUBCO(I)
 3600   CONTINUE
C
        SSUM(21)=SSUM(21)+PCOST
        SSUMSQ(21)=SSUMSQ(21)+PCOST*PCOST
        SUMXY(11)=SUMXY(11)+KEYDUR(NKEYEN)*PCOST
C
        RETURN
        END
C------------------------------------------------------------------
C
C       SUBROUTINE      S  S  Z
C
C       THIS  SUBROUTINE    CALCULATES    THE   SAMPLE   SIZE   USING   A  LINEAR   INTER-
C       POLATION    BETWEEN    ENDPOINTS    (SEE   SAMPLE   SIZE   INFO.   FOR   MORE)
C
C------------------------------------------------------------------
C
        SUBROUTINE     SSZ(X1,X2,Y1,Y2,RAL,VAL)
        REAL   X1,X2,Y1,Y2,RAL,VAL
C
        SLOPE=(Y2-Y1)/(X2-X1)
        B=Y1-SLOPE*X1
C
        VAL=(1-RAL-B)/SLOPE
C
        RETURN
        END

C------------------------------------------------------------------
C
C       SUBROUTINE      S  T  D  N  O  R
C
C       THIS  SUBROUTINE    COMPUTES    THE   Pr  ( X  <=  x )  FOR  A  STANDARD
C       NORMAL   VARIABLE.
C       THE  PROBABILITY    COMPUTED    IS  DENOTED   BY  PHIZ.
C
```

```
C-----------------------------------------------------------------
C
      SUBROUTINE    STDNOR(Z,PHIZ)
C
      REAL   Z,PHIZ
      DOUBLE   PRECISION      PI
      DATA   PI/3.141592653589793/
C
      Y=Z/SQRT(2.0)
      IF  (Z.LE.0)    Y=-Y
      S=0
C
      DO  100  I=1,37
        RI=I
        S=S+EXP( RI*RI/25.)/RI*SIN(2*RI*Y/5.)
  100 CONTINUE
C
      S=S+Y/5.
      S=2*S/PI
      PHIZ=(1+S)/2.
C
      IF  (Z.GE.0.)    GOTO   1780
      PHIZ=(1-S)/2.
 1780 IF  (Z.LE.8.3)     GOTO   1800
      PHIZ=1
 1800 IF  (Z.GE.-8.3)     GOTO   1820
      PHIZ=0
C
 1820 RETURN
      END
C-----------------------------------------------------------------
C
C         SUBROUTINE      S  U  B  N  E  T
C
C     THIS   SUBROUTINE     FINDS   THE   SUBNETWORKS    OF   THE   KEY  EVENTS    AND
C   ***************************COSTS              TO   THE   KEY   EVENTS.
C
C-----------------------------------------------------------------
C
      SUBROUTINE     SUBNET(NKEYEN,NACT,KEYEVN,NODEFI,NODEST,ACOST,ASCEND,
     *                          SUBCO,SUBM,PERCEN,FLAGONE)
C
      INTEGER     NKEYEN,NACT,KEYEVN(10),NODEFI(100),NODEST(100),
     *            ASCEND(100),FLAGONE
      REAL        ACOST(100),SUBCO(11),SUBM(100,10),PERCEN(100,10)
C
      INTEGER     LP(100)
      REAL        SUBN(100,10)
C
C
C---INITIALIZE      SUBNETWORK     MATRIX    TO  ZERO
C
      DO  1001   K=1,NKEYEN
```

```
      DO  1000  I=1,NACT
        SUBN(I,K)=0.0
 1000 CONTINUE
 1001 CONTINUE
C
C---FIND  SUBNETWORKS  OF  THE  KEY  EVENTS  AND  STORE  IN  SUBN
C
      DO  1005  K=1,NKEYEN
        LZ  =  KEYEVN(K)
        DO  1010  I=1,NACT
          LP(I)=0
 1010   CONTINUE
 1100   KT=0
        DO  1015  I=1,NACT
          IF  (NODEFI(I).NE.LZ)     GO  TO  1015
          IF  (KT.EQ.0)    GO  TO  1105
          LP(I)=1
          GO  TO  1015
 1105     SUBN(I,K)=ACOST(I)
          KT=1
          LE=NODEST(I)
 1015   CONTINUE
        LZ=LE
        IF  (LZ.NE.ASCEND(1))     GO  TO  1100
 1110   DO  1020  I=1,NACT
          IF  (LP(I).NE.0)    GO  TO  1115
 1020   CONTINUE
        GO  TO  1005
 1115   KS=0
        DO  1025  I=1,NACT
          IF  (LP(I).EQ.0)    GO  TO  1025
          IF  (KS.EQ.1)    GO  TO  1025
          SUBN(I,K)=ACOST(I)
          LP(I)=0
          LZ=NODEST(I)
          KS=1
 1025   CONTINUE
        GO  TO  1100
 1005 CONTINUE
C
C---UPDATE  SUBGRAPHS  BY  REMOVING  ACTIVITIES  IN  KEY  EVENTS  THAT
C---HAVE  KEY  EVENTS  AS  SUBSETS  (STORE  IN  SUBM)
C
      DO  1050  I=1,NACT
      DO  1050  K=1,NKEYEN
        SUBM(I,K)=SUBN(I,K)
 1050 CONTINUE
C
      MKEYEN=NKEYEN-1
      DO  1055  J=1,MKEYEN
        L=J+1
        DO  1056  K=L,NKEYEN
        DO  1057  I=1,NACT
```

```
         IF  (SUBN(I,J).GT.SUBN(I,K))          GO  TO  1056
 1057      CONTINUE
           DO  1058   I=1,NACT
             IF  (SUBN(I,J).EQ.0)       GO  TO  1058
             SUBM(I,K)=0.
 1058      CONTINUE
 1056      CONTINUE
 1055    CONTINUE
C
C----------------------------------------------------------------
C
C----ON   INITIAL   CALL,   I  NEED   TO  RETURN   HERE----
C
         IF  (FLAGONE.EQ.0)       RETURN
C--------------------------------------------
C
         DO  1070   I=1,NACT
           TOTAL=0.
         DO  1075   K=1,NKEYEN
           TOTAL=TOTAL+SUBM(I,K)
 1075    CONTINUE
C
C---IF   THE  ACTIVITY   IS  COMMON   TO MULTIPLE   KEY  EVENTS,   WE  NOW  SPLIT
C---THE   COST  ACCORDING   TO  OUR  PREVIOUS   DECISION
C
         IF  (TOTAL.GT.ACOST(I))       THEN
            DO  1077   K=1,NKEYEN
             SUBM(I,K)=SUBM(I,K)*PERCEN(I,K)
 1077       CONTINUE
         ENDIF
 1070    CONTINUE
C
         WRITE(*,*)     'COST   PERCENTAGES     TO  KEY   EVENTS:'
         WRITE(*,*)     ' '
         WRITE(*,1090)      (K,K=1,NKEYEN)
 1090    FORMAT(/4X,'NS          NF',10('       KEY(',I2,')')/)
         DO  1076   I=1,NACT
           WRITE(*,1091)     NODEST(I),NODEFI(I),(SUBM(I,K),K=1,NKEYEN)
 1091    FORMAT(1X,2I5,10(F10.2))
 1076    CONTINUE
C
C
 1081    RETURN
         END
C-----------------------------------------------------------------
C
C        SUBROUTINE      U  P  P  A  Y
C
C           UPDATE   ARRAY   PAY   AND   VAR.   CAPITL   CONTAINING    THE   AMOUNT   OF
C        COST   OBTAINED    IN  THE   PERIOD   OF  THE   KEY   EVENT'S    REALIZATION.
C
C-----------------------------------------------------------------
C
```

```
      SUBROUTINE    UPPAY(KEYDUR,PAY,MXPERD,CYCLE,NMCS,NKEYEN,SUBCO,CAPITL)
C
      INTEGER   NKEYEN,MXPERD,CYCLE,NMCS
      REAL      KEYDUR(10),PAY(10,100),CAPITL
      DOUBLE    PRECISION    SUBCO(11)
C
      REAL      TPAY(10,100),TDURAT
C
C----INITIALIZE      ARRAY    TPAY
C
      DO  1204   K1=1,NKEYEN
      DO  1205   K2=1,MXPERD
        TPAY(K1,K2)=0.
 1205 CONTINUE
 1204 CONTINUE
C
      DO  1200   K2=1,NKEYEN
        IMARK=KEYDUR(K2)/CYCLE
        RMARK=KEYDUR(K2)/CYCLE
        IF  ((RMARK-IMARK).EQ.0.00)         THEN
          TPAY(K2,IMARK)=TPAY(K2,IMARK)+SUBCO(K2)
        ELSE
          TPAY(K2,IMARK+1)=TPAY(K2,IMARK+1)+SUBCO(K2)
        ENDIF
 1200 CONTINUE
C
C----UPDATE     VALUES    FOR   MONTE    CARLO    SIM.----
C
      CAPITL=CAPITL+SUBCO(NKEYEN+1)/NMCS
      DO  1209   K2=1,NKEYEN
      DO  1210   K1=1,MXPERD
        PAY(K2,K1)=PAY(K2,K1)+TPAY(K2,K1)/NMCS
 1210 CONTINUE
 1209 CONTINUE
C
      RETURN
      END
C---------------------------------------------------------------
C
C        SUBROUTINE      X E X P E C
C
C      THIS   SUBROUTINE    DETERMINES    THE   MEAN   AND   VARIANCE    OF   THE
C      ACTIVITIES     FOR   USE   IN   SUBROUTINE     INITIALIZE.
C
C---------------------------------------------------------------
C
      SUBROUTINE     XEXPEC(XEXP,XSTD,NACT,P1,P2,P3,P4,DNUM,DIST,DURA,NOP)
C
      INTEGER    DNUM(100),NOP(100),NACT
      REAL   P1(100),P2(100),P3(100),P4(100),DIST(100,30),DURA(100,30),
     *       XEXP(100),XSTD(100)
C
      DO  1950   I=1,NACT
```

```
      Y1=P1(I)
      Y2=P2(I)
      Y3=P3(I)
      Y4=P4(I)
      Y5=Y4-Y3
      GOTO  (1952,1953,1954,1955,1956,1957,1958),DNUM(I)
C
C ---  UNIFORM
C
 1952 X=(Y3+Y4)/2.0
      ST=SQRT(Y5*Y5/12.0)
      GOTO  1959
C
C ---  TRIANGULAR
C
 1953 X=(Y1+Y3+Y4)/3.0
      ST=SQRT((Y3*(Y3-Y1)+Y4*Y5+Y1*(Y1-Y4))/18.0)
      GOTO  1959
C
C ---  NORMAL
C
 1954 X=Y1
      ST=Y2
      GOTO  1959
C
C ---  EXPONENTIAL
C
 1955 X=Y1
      ST=Y1
      GOTO  1959
C
C ---  GAMMA
C
 1956 X=Y1*Y2
      ST=SQRT(X*Y2)
      GOTO  1959
C
C ---  BETA
C
 1957 X=Y3+Y5*Y1/(Y1+Y2)
      ST=SQRT(Y5*Y5*Y1*Y2/((Y1+Y2)**2*(Y1+Y2+1)))
      GOTO  1959
C
C ---  DISCRETE
C
 1958 X=0
      X2=0
      DO  1951  I1=1,NOP(I)
        X=X+DIST(I,I1)*DURA(I,I1)
        X2=X2+DIST(I,I1)*DURA(I,I1)**2
 1951 CONTINUE
      ST=SQRT(X2-X*X)
C
```

```
1959     XEXP(I)=X
         XSTD(I)=ST
1950  CONTINUE
      RETURN
      END
```

Appendix 9.4. Program Listing of CASH

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C     PROGRAM     C A S H                          31 NOV 86          C
C                                                                     C
C     WRITTEN BY RUSSELL S. VOGTMANN                                  C
C                                                                     C
C     THIS PROGRAM ALLOWS THE USER TO USE THE INFORMATION OBTAINED FROM C
C     THE MONTE CARLO SIMULATION FROM PROGRAM AN-COST TO DETERMINE A  C
C     BID PACKAGE FOR THE PROJECT.                                    C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C    VARIABLES:
C
C    ALPHA:           INTEREST RATE ON MONEY DEPOSITED.
C    BETA:            INTEREST RATE ON MONEY BORROWED.
C    BHAT0(  ):       CONTAINS INTERCEPT TERM OF THE SAMPLE REGRESSION
C                     FUNCTION OF KEY EVENT COST AND DURATION.
C    BHAT1(  ):       CONTAINS COEFFICIENT OF DURATION TERM OF THE SAMPLE
C                     REGRESSION FUNCTION.
C    CAPITAL:         CONTRACTOR'S INITIAL CAPITAL.
C    CDF( , ):        EMPIRICAL CUMULATIVE DISTRIBUTION FUNCTION OF DURATION
C                     AND COST OF KEY EVENTS.
C    CMODEL:          CASH FLOW MODEL IN DETERMINISTIC PROBLEM --- 1. PROFIT=
C                     f(CAPITAL), 2. PROFIT=f(PROJECT COST).
C    CORR(  ):        SAMPLE CORRELATION COEFFICIENT OF COST vs. DURATION OF
C                     KEY EVENTS.
C    COST(  ):        USER DECIDED COST TO RECEIVE AT KEY EVENTS FOR PROB-
C                     ABILISTIC CASH FLOW PROBLEM.
C    DF( , ):         EMPIRICAL DENSITY FUNCTION OF DURATION AND COST OF KEY
C                     EVENTS.
C    DMEAN(  ):       MEAN OF DURATION OF KEY EVENTS.
C    DSTD(  ):        STANDARD DEVIATION OF DURATION OF KEY EVENTS.
C    DURN(  ):        USER DECIDED DUE DATE OF KEY EVENTS FOR PROBABILISTIC
C                     CASH FLOW PROBLEM.
C    DESIRE:          DESIRED PROFIT PERCENTAGE FOR DETERMINISTIC CASH FLOW
C                     PROBLEM.
C    FRACN(  ):       EXPECTED FRACTION OF TOTAL PROJECT COST FOR EACH KEY
C                     EVENT (AFTER START PERCENTAGE<=>STPERC IS REMOVED).
C    IKEY(  ):        ARRAY CONTAINING THE NODE NUMBERS OF THE KEY EVENTS.
C    LL(  ):          LEFT ENDPOINTS OF KEY EVENT DISTRIBUTIONS.
C    MXPERD:          MAXIMUM NUMBER OF PERIODS THAT PROJECT RUNS OVER.
C    NCELLS(  ):      NUMBER OF CELLS OF KEY EVENT DISTRIBUTIONS.
C    NCYCLE:          NUMBER OF TIME UNITS PER PERIOD.
C    NKEYEN:          NUMBER OF KEY EVENTS.
C    NRR(  ):         RIGHT ENDPOINTS OF KEY EVENT DISTRIBUTIONS.
C    PAY( , ):        EXPECTED CASH "INFLOW" FOR EACH KEY EVENT PER PERIOD
C                     (DETERMINED BY AN-COST).
C    PENALTY(  ):     USER DECIDED PENALTY PER PERIOD LATE FOR PROBABILISTIC
C                     CASH FLOW PROBLEMS.
C    PERCNT( , ):     PERCENTILES OF KEY EVENT DISTRIBUTIONS.
```

```
      OPEN(UNIT=2,FILE='CASHFLOW.DAT',STATUS='OLD')
      OPEN(UNIT=3,FILE='CASHOUT.OUT',STATUS='NEW')
C
      IFLAGG=0
      WRITE(*,1)
    1 FORMAT(25(/))
      WRITE(*,2)
    2 FORMAT(/20X,'          PROGRAM                    CASH',//////,
     &20X,'                  WRITTEN    BY',/////,
     &20X,'      RUSSELL        S.       VOGTMANN',///)
      PAUSE
      WRITE(*,1)
C -----------------------------------------------------------------
C       INPUT    VALUES    FROM    FILE    CASHFLOW.DAT
C -----------------------------------------------------------------
      WRITE(*,*)     'INPUTTING     VALUES     FROM     FILE     CASHFLOW.DAT'
      READ(2,90)     NCYCLE
      READ(2,90)     MXPERD
      READ(2,91)     (SDISBUR(I),I=1,MXPERD)
   90 FORMAT(10I5)
   91 FORMAT(5E13.7)
      READ(2,90)     NKEYEN
      READ(2,90)     (IKEY(I),I=1,NKEYEN)
      DO  10  I=1,NKEYEN
        READ(2,91)     (PAY(I,J),J=1,MXPERD)
   10 CONTINUE
      READ(2,91)     STPERC,PINIT
      DO  11  I=1,NKEYEN
        READ(2,91)     DMEAN(I),DSTD(I)
        READ(2,91)     RMEAN(I),RSTD(I)
   11 CONTINUE
      READ(2,91)     RMEAN(11),RSTD(11)
      DO  12  I=1,NKEYEN
        READ(2,92)     LL(I),NRR(I),NCELLS(I),WIDTH(I)
        READ(2,91)     (DF(I,J),J=1,(NCELLS(I)+2))
        READ(2,92)     LL(I+10),NRR(I+10),NCELLS(I+10),WIDTH(I+10)
        READ(2,91)     (DF(I+10,J),J=1,(NCELLS(I+10)+2))
   12 CONTINUE
      READ(2,92)     LL(22),NRR(22),NCELLS(22),WIDTH(22)
      READ(2,91)     (DF(22,J),J=1,(NCELLS(22)+2))
   92 FORMAT(I10,I10,I10,F15.5)
      DO  13  I=1,NKEYEN
        READ(2,91)     CORR(I),BHATO(I),BHAT1(I)
   13 CONTINUE
      READ(2,91)     CORR(11),BHATO(11),BHAT1(11)
      DO  14  I=1,NKEYEN
        READ(2,91)     FRACN(I)
   14 CONTINUE
C-----------------------------------------------------------------
      DO  300  I=1,10
        COST(I)=0.0
        DURN(I)=0.0
        PENALTY(I)=0.0
```

```
  300  CONTINUE
C
C---DETERMINE    THE  EXPECTED   TOTAL  CASH  "INFLOW"   PER  PERIOD
C
  240  DO 244 I=1,100
         TPAY(I)=0.0
  244  CONTINUE
C
       DO 242 I=1,MXPERD
       DO 241 J=1,NKEYEN
         TPAY(I)=TPAY(I)+PAY(J,I)
  241  CONTINUE
  242  CONTINUE
C
C---CALCULATE    CDF'S  AND  STORE  IN  CDF
C
       CALL CUMDIST
       WRITE(*,1)
C
 1000  WRITE(*,4)
    4  FORMAT(/1X,'INPUT      OPTION   YOU  WANT  TO  TAKE:',//,
     &10X,'1.     VIEW  CASH  FLOW  ARRAYS',/,
     &10X,'2.     VIEW  DISTRIBUTION    FUNCTION    INFORMATION',/,
     &10X,'3.     PERFORM   DETERMINISTIC     CASH  FLOW  PROBLEM',/,
     &10X,'4.     PERFORM   PROBABILISTIC     CASH  FLOW  PROBLEM',/,
     &10X,'5.     PROGRAM   INFORMATION',/,
     &10X,'6.     -------------EXIT      PROGRAM-------------',///)
       READ(*,*)    NOPTION
       IF  (NOPTION.LT.1.OR.NOPTION.GT.6)          THEN
         WRITE(*,1)
         GOTO  1000
       ENDIF
C
       IF  (NOPTION.EQ.1)     CALL   VIEW
       IF  (NOPTION.EQ.2)     CALL   DISTFN(IFLAGG)
       IF  (NOPTION.EQ.3)     CALL   DETERM(NOPTION)
       IF  (NOPTION.EQ.4)     CALL   PROBAB(NOPTION,IFLAGG)
       IF  (NOPTION.EQ.5)     CALL   HELPMAIN
       IF  (NOPTION.EQ.6)     STOP
       WRITE(*,1)
       GOTO  1000
C
       STOP
       END
C-----------------------------------------------------------------
C
C      SUBROUTINE     C F H E L P
C
C      THIS  SUBROUTINE    PROVIDES    INFORMATION    ON  THE  CASH  FLOW  ARRAYS
C
C-----------------------------------------------------------------
C
       SUBROUTINE    CFHELP
```

```
C
      WRITE(*,1)
    1 FORMAT(25(/))
      WRITE(*,2)
    2 FORMAT(5X,'THE    EXPECTED   CASH   FLOW   PER   PERIOD   IS   CALCULATED    BY',/,
     *5X,'PROGRAM    AN-COST    BY   MEANS   OF   MONTE   CARLO   SIMULATION',//,
     *5X,'IF    THE   USER   HAS   DETERMINED    A  PERCENTAGE    OF   COST   TO   RECEIVE',/,
     *5X,'AT    THE   PROJECT    START,    IT   IS  DENOTED    AS  OCCURRING    IN   PERIOD',/,
     *5X,'ZERO.',/////)
      PAUSE
C
      RETURN
      END
C------------------------------------------------------------------
C
C       SUBROUTINE      C  F  L  O  W
C
C       THIS   SUBROUTINE    PERFORM    CASH   FLOW   CALCULATIONS
C
C------------------------------------------------------------------
C
      SUBROUTINE    CFLOW(PROFIT,RATIO,CASH,TCAPITAL,SDISBUR,PAY,ALPHA,
     *                    BETA,RATE,DESI,POWER,MXPERD,FLAG2,CTERM,
     *                    TCAPINIT,CMODEL,RMEAN,TPAY)
C
      INTEGER    MXPERD,FLAG2,CMODEL
      REAL       RATIO,CASH,TCAPITAL,SDISBUR(100),PAY(10,100),ALPHA,BETA,
     *           RATE,POWER,DESI,PROFIT,CTERM,TCAPINIT,RMEAN(11),TPAY(100)
C
      REAL       TIAF,TIBA,TRETEN,RETEN
C
      IF  (FLAG2.EQ.1)     THEN
        WRITE(3,1900)      RATIO
 1900   FORMAT(/3X,'RATIO       (ACTIVITY     WORTH    /  ACTIVITY    COST    )  =',F8.4)
      ENDIF
C
        TIAF=0.00
        TIBA=0.00
        RETEN=0.00
        TRETEN=0.00
      DO  1510    I=1,MXPERD+1
C
C----------------------
C
        IF  (I.EQ.1)    THEN
          CASH=TCAPITAL-SDISBUR(I)
          CALL   INTRT(I,CASH,POWER,ALPHA,BETA,TIAF,TIBA,TRETEN,FLAG2)
          GO  TO  1510
        ENDIF
C
C----------------------
C
        IF  (I.EQ.MXPERD+1)      THEN
```

```
        CASH=CASH+TPAY(I-1)+TRETEN
        CTERM=CASH
        IF (FLAG2.EQ.1) THEN
          WRITE(3,1991) CASH
 1991     FORMAT(25X,'TERMINAL CASH POSITION =',F10.3)
        ENDIF
        GO TO 1510
      END IF
C
C-------------------------
C
        RETEN=TPAY(I-1)*RATIO*RATE
        TRETEN=TRETEN+RETEN
        CASH=CASH+TPAY(I-1)*RATIO*(1-RATE)-SDISBUR(I)
        CALL INTRT(I,CASH,POWER,ALPHA,BETA,TIAF,TIBA,TRETEN,FLAG2)
        IF (FLAG2.EQ.1) THEN
          WRITE(3,1980) TRETEN,RETEN
 1980     FORMAT(/15X,'CUMULATIVE RETENTION =',F10.3,3X,'RETENTION=',
     *           F9.3/)
        ENDIF
C
C-------------------------
C
 1510 CONTINUE
        IF (CMODEL.EQ.1) PROFIT=(CASH-TCAPINIT)/TCAPINIT
        IF (CMODEL.EQ.2) PROFIT=(CASH-TCAPINIT)/RMEAN(11)
        IF (FLAG2.EQ.1) THEN
          WRITE(3,1995) PROFIT
 1995     FORMAT(20X,'PROFIT =',F10.3/)
        ENDIF
C
 1500 CONTINUE
C
      RETURN
      END
C-------------------------------------------------------------------
C
C     FUNCTION C U M (JJ,I)
C
C     THIS FUNCTION CALCULATES THE PROBABILITY THAT THE KEY EVENT JJ
C     IS REALIZED IN PERIOD I. IT USES A LINEAR INTERPOLATION.
C
C-------------------------------------------------------------------
C
      REAL FUNCTION CUM(NEVENT,PERIOD)
C
      COMMON/CFMV/SDISBUR(100),PAY(10,100),RMEAN(11),PINIT,MXPERD,NKEYEN
     *,IKEY(10),TPAY(100),DMEAN(10),DF(22,102),LL(22),NRR(22),NCELLS(22)
     *,WIDTH(22),DSTD(10),RSTD(11),CDF(22,102),CORR(11),BHAT0(11)
     *,BHAT1(11),FRACN(10)
      COMMON/CFP/ALPHA,BETA,POWER,RATE,DESIRE,NCYCLE,CAPITAL,TOL,CMODEL
      COMMON/DCV/COST(11),DURN(10),STPERC,PENALTY(10),PERCNT(22,20)
      INTEGER NEVENT,PERIOD,RLEFT,RIGHT
```

```
      DOUBLE    PRECISION     V1,V2,SLOPE,B
C
      RLEFT=NCYCLE*(PERIOD-1)
      RIGHT=NCYCLE*PERIOD
C
      CUM=0.0
      V1=0.0
      V2=0.0
      SLOPE=0.0
      B=0.0
C
      IF  (RLEFT.LT.LL(NEVENT)-WIDTH(NEVENT))             THEN
        V1=0.0
        GOTO   4100
      ENDIF
      IF  (RLEFT.GT.NRR(NEVENT)+WIDTH(NEVENT))             THEN
        V1=1.0
        GOTO   4100
      ENDIF
C
      DO  4000   K=1,NCELLS(NEVENT)+2
        P1=LL(NEVENT)+WIDTH(NEVENT)*(K-2)
        P2=LL(NEVENT)+WIDTH(NEVENT)*(K-1)
        IF  (RLEFT.GT.P1.AND.RLEFT.LE.P2)            THEN
          IF  (K.EQ.1)    THEN
            SLOPE=(CDF(NEVENT,1))/WIDTH(NEVENT)
          ELSE
            SLOPE=(CDF(NEVENT,K)-CDF(NEVENT,K-1))/WIDTH(NEVENT)
          ENDIF
          B=CDF(NEVENT,K)-SLOPE*P2
          V1=SLOPE*RLEFT+B
          GOTO   4100
        ENDIF
 4000 CONTINUE
C
 4100 IF  (RIGHT.LT.LL(NEVENT)-WIDTH(NEVENT))             THEN
        V2=0.0
        GOTO   4300
      ENDIF
      IF  (RIGHT.GT.NRR(NEVENT)+WIDTH(NEVENT))             THEN
        V2=1.0
        GOTO   4300
      ENDIF
C
      DO  4200   K=1,NCELLS(NEVENT)+2
        P1=LL(NEVENT)+WIDTH(NEVENT)*(K-2)
        P2=LL(NEVENT)+WIDTH(NEVENT)*(K-1)
        IF  (RIGHT.GT.P1.AND.RIGHT.LE.P2)            THEN
          IF  (K.EQ.1)    THEN
            SLOPE=CDF(NEVENT,1)/WIDTH(NEVENT)
          ELSE
            SLOPE=(CDF(NEVENT,K)-CDF(NEVENT,K-1))/WIDTH(NEVENT)
          ENDIF
```

```
            B=CDF(NEVENT,K)-SLOPE*P2
            V2=SLOPE*RIGHT+B
            GOTO   4300
          ENDIF
 4200   CONTINUE
C
 4300   CUM=V2-V1
        RETURN
        END
C-----------------------------------------------------------------
C
C       SUBROUTINE      C U M D I S T
C
C       THIS   SUBROUTINE    CALCULATES    THE   CUMULATIVE    DIST.    FUNC.'S
C
C-----------------------------------------------------------------
C
        SUBROUTINE    CUMDIST
C
        COMMON/CFMV/SDISBUR(100),PAY(10,100),RMEAN(11),PINIT,MXPERD,NKEYEN
       *,IKEY(10),TPAY(100),DMEAN(10),DF(22,102),LL(22),NRR(22),NCELLS(22)
       *,WIDTH(22),DSTD(10),RSTD(11),CDF(22,102),CORR(11),BHAT0(11)
       *,BHAT1(11),FRACN(10)
        DOUBLE    PRECISION    SUM
C
        MARK1=1
        MARK2=0
C
 3000   DO  3002   I=MARK1,NKEYEN+MARK2
          SUM=0.
          DO  3005   J=1,NCELLS(I)+2
            SUM=SUM+DF(I,J)
            CDF(I,J)=SUM
 3005     CONTINUE
 3002   CONTINUE
C
        IF  (MARK1.EQ.1)      THEN
          MARK1=11
          MARK2=10
          GOTO   3000
        ENDIF
C
        IF  (MARK1.EQ.11)      THEN
          MARK1=22
          MARK2=22-NKEYEN
          GOTO   3000
        ENDIF
C
        RETURN
        END
C-----------------------------------------------------------------
C
C       SUBROUTINE      D C F H E L P
```

```
C
C      THIS SUBROUTINE PROVIDES INFORMATION ON THE CASH FLOW CALCULATIONS
C      IN THE CASE OF A DETERMINISTIC NETWORK.
C
C-----------------------------------------------------------------
C
       SUBROUTINE DCFHELP
C
       WRITE(*,1)
     1 FORMAT(25(/))
       WRITE(*,2)
     2 FORMAT(5X,'THIS PORTION OF THE PROGRAM DOES CASH FLOW CALCULA-',/,
      *5X,'TIONS IN THE CASE OF A DETERMINISTIC NETWORK.',//,
      *5X,'YOU WILL BE ASKED TO INPUT:',//,
      *10X,'1. INTEREST RATE ON MONEY DEPOSITED',/,
      *10X,'2. INTEREST RATE ON MONEY BORROWED',/,
      *10X,'3. RETENTION RATE',/,
      *10X,'4. INITIAL CAPITAL',//,
      *10X,'5. DESIRED PROFIT PERCENTAGE',/,
      *10X,'6. IF PROFIT IS DEFINED AS A FUNCTION OF PROJECT COST,',/,
      *10X,'   OR AS A FUNCTION OF CAPITAL',/,
      *10X,'7. A TOLERANCE LEVEL FOR USE IN THE BISECTION METHOD',//)
       PAUSE
C
       WRITE(*,10)
    10 FORMAT(5(/))
       WRITE(*,3)
     3 FORMAT(5X,'THIS CALCULATION PROCEDURE USES THE BISECTION',/,
      *5X,'METHOD TO CONVERGE TO THE MARKUP OF ACTIVITY COST THAT',/,
      *5X,'WILL RESULT IN THE DESIRED PROFIT PERCENTAGE.',///,
      *5X,'YOU MAY RUN AS MANY ALTERNATIVES AS YOU LIKE. THE RESULTS',/,
      *5X,'WILL BE SENT TO THE SCREEN, AND TO AN OUTPUT FILE NAMED',/,
      *5X,'"CASHOUT.OUT" RESIDING ON THE DEFAULT DRIVE.',///)
       PAUSE
C
       RETURN
       END
C-----------------------------------------------------------------
C
C      SUBROUTINE   D E T E R M
C
C      THIS SUBROUTINE DOES CASH FLOW CALCULATIONS FOR DETERMINISTIC
C      PROBLEMS AND CALCULATES A FaRM. IT CONVERGES TO THE VALUE USING
C      THE BISECTION METHOD.
C
C-----------------------------------------------------------------
C
       SUBROUTINE DETERM(NOPTION)
C
       COMMON/CFMV/SDISBUR(100),PAY(10,100),RMEAN(11),PINIT,MXPERD,NKEYEN
      *,IKEY(10),TPAY(100),DMEAN(10),DF(22,102),LL(22),NRR(22),NCELLS(22)
      *,WIDTH(22),DSTD(10),RSTD(11),CDF(22,102),CORR(11),BHAT0(11)
      *,BHAT1(11),FRACN(10)
```

```
      COMMON/CFP/ALPHA,BETA,POWER,RATE,DESIRE,NCYCLE,CAPITAL,TOL,CMODEL
      INTEGER   FLAG2,CMODEL
C
 1549 WRITE(*,1550)
 1550 FORMAT(25(/))
      WRITE(*,1551)
 1551 FORMAT(////10X,'1.        PROCEED   WITH   CASH   FLOW   CALCULATIONS',/,
     *10X,'2.     DETERMINISTIC      CASH   FLOW   INFORMATION',/,
     *10X,'3.     RETURN   TO  MAIN   MENU',///)
      WRITE(*,1552)
 1552 FORMAT(5X,'ENTER       CHOICE')
      READ(*,*)     ICHOICE
      IF  (ICHOICE.EQ.2)      THEN
        CALL   DCFHELP
        GOTO   1549
      ENDIF
      IF  (ICHOICE.EQ.3)      RETURN
C
      CALL   INPUT(NOPTION,ISTOP,IFLAGG)
C
      POWER=NCYCLE/365.0
      DESI=DESIRE/100.0
C
C---THE    FOLLOWING    IS  A  TESTING    OF  VALUES    TO  GET   A  RIGHT   END   STARTING
C---POINT     FOR   THE   BISECTION     METHOD
C
      TCAPINIT=CAPITAL
      TCAPITAL=CAPITAL+PINIT
      FLAG2=0
      TESTVAL=2
  520 CALL    CFLOW(PROFIT,TESTVAL,CASH,TCAPITAL,SDISBUR,PAY,ALPHA,BETA,
     *              RATE,DESI,POWER,MXPERD,FLAG2,CTERM,TCAPINIT,CMODEL,
     *              RMEAN,TPAY)
C
      IF  (PROFIT.GT.DESI)       THEN
        RATIO2=TESTVAL
        GO  TO  530
      ELSE
        TESTVAL=TESTVAL+0.5
        G   TO  520
      ENDIF
C
C---NOW    I  USE   BISECTION     TO  CONVERGE    TO  THE   FaRM
C
  530 RATIO1=0.50
      CALL    CFLOW(PROFT1,RATIO1,CASH,TCAPITAL,SDISBUR,PAY,ALPHA,BETA,
     *                  RATE,DESI,POWER,MXPERD,FLAG2,CTERM,TCAPINIT,
     *                  CMODEL,RMEAN,TPAY)
C
C---THE    FOLLOWING    IS  BASICALLY    A  WHILE   LOOP   FOR   THE   BISECTION
C
  600 RATIO=RATIO1+(RATIO2-RATIO1)/2.0
C
```

```
      CALL   CFLOW(PROFIT,RATIO,CASH,TCAPITAL,SDISBUR,PAY,ALPHA,BETA,
     *              RATE,DESI,POWER,MXPERD,FLAG2,CTERM,TCAPINIT,CMODEL,
     *              RMEAN,TPAY)
C
      IF  (ABS(PROFIT-DESI).LT.TOL)          THEN
        CALL   DETOUT(NOPTION,RATIO)
        RETURN
      ENDIF
C
      IF  ((DESI-PROFT1)*(DESI-PROFIT).LT.0.00)               THEN
        RATIO2=RATIO
      ELSE
        RATIO1=RATIO
        PROFT1=PROFIT
      ENDIF
C
      GO  TO  600
C
      RETURN
      END
C----------------------------------------------------------------------
C
C     SUBROUTINE      D E T O U T
C
C     THIS   SUBROUTINE    OUTPUTS    DATA   TO  FILE   "CASHOUT.OUT"
C
C----------------------------------------------------------------------
C
      SUBROUTINE     DETOUT(NOPTION,RATIO)
C
      COMMON/CFMV/SDISBUR(100),PAY(10,100),RMEAN(11),PINIT,MXPERD,NKEYEN
     *,IKEY(10),TPAY(100),DMEAN(10),DF(22,102),LL(22),NRR(22),NCELLS(22)
     *,WIDTH(22),DSTD(10),RSTD(11),CDF(22,102),CORR(11),BHAT0(11)
     *,BHAT1(11),FRACN(10)
      COMMON/CFP/ALPHA,BETA,POWER,RATE,DESIRE,NCYCLE,CAPITAL,TOL,CMODEL
      REAL   RATIO
      INTEGER    FLAG2,NOPTION,CMODEL
C
 1555 FLAG2=1
      TCAPINIT=CAPITAL
      TCAPITAL=CAPITAL+PINIT
      CALL   CFLOW(PROFIT,RATIO,CASH,TCAPITAL,SDISBUR,PAY,ALPHA,BETA,
     *              RATE,DESI,POWER,MXPERD,FLAG2,CTERM,TCAPINIT,CMODEL,
     *              RMEAN,TPAY)
      TWORTH=RMEAN(11)*RATIO
      WRITE(3,1900)
      WRITE(*,1900)
 1900 FORMAT(10X,'*********DETERMINISTIC             RESULTS*********',//)
      WRITE(3,1901)      DESIRE
      WRITE(*,1901)      DESIRE
 1901 FORMAT(2X,'IN      ORDER    TO   MAKE',F6.2,'%      PROFIT    IN   THIS    PROJECT')
      WRITE(3,1902)      RATIO
      WRITE(*,1902)      RATIO
```

```
 1902   FORMAT(/2X,'THE       BIDDING    SHOULD    BE',F8.4,'       TIMES    OF   THE')
        WRITE(3,1903)       RMEAN(11),TWORTH
        WRITE(*,1903)       RMEAN(11),TWORTH
 1903   FORMAT(/2X,'PROJECT          COST    =',F12.4,'      THAT   IS',F12.4,)
        WRITE(3,1904)
        WRITE(*,1904)
 1904   FORMAT(/2X,'UNDER         THE   CONSTRAINTS:       ')
        WRITE(3,1905)       CAPITAL
        WRITE(*,1905)       CAPITAL
 1905   FORMAT(/8X,'INITIAL          CAPITAL     =',F10.2)
        WRITE(3,1906)       ALPHA
        WRITE(3,1907)       BETA
        WRITE(3,1908)       RATE
        WRITE(*,1906)       ALPHA
        WRITE(*,1907)       BETA
        WRITE(*,1908)       RATE
 1906   FORMAT(/8X,'INTEREST         RATE   ON   MONEY    DEPOSIT     =',F6.4)
 1907   FORMAT(/8X,'INTEREST         RATE   ON   MONEY    BORROWT     =',F6.4)
 1908   FORMAT(/8X,'RETENTION         RATE   =  ',F6.4)
C
        RETURN
        END
C-----------------------------------------------------------------------
C
C       SUBROUTINE       D  I  S  T  F  N
C
C       THIS   SUBROUTINE    ALLOWS    THE   USER   TO  VIEW   THE   DISTRIBUTION     FUNC.'S
C       OF  DURATION     AND   COST    OF   THE    KEY   EVENTS    OF   THE   NETWORK.     IT  ALSO
C       WILL   DISPLAY    THE   CORRELATION     COEFFICIENT     BETWEEN    THE   DURATION    AND
C       COST   OF   THE   KEY   EVENTS,    AS   WELL   AS  DETERMINE     PERCENTILES     OF   THE
C       DISTRIBUTIONS.
C
C-----------------------------------------------------------------------
C
        SUBROUTINE     DISTFN(IFLAGG)
C
        COMMON/CFMV/SDISBUR(100),PAY(10,100),RMEAN(11),PINIT,MXPERD,NKEYEN
      *,IKEY(10),TPAY(100),DMEAN(10),DF(22,102),LL(22),NRR(22),NCELLS(22)
      *,WIDTH(22),DSTD(10),RSTD(11),CDF(22,102),CORR(11),BHAT0(11)
      *,BHAT1(11),FRACN(10)
        COMMON/DCV/COST(11),DURN(10),STPERC,PENALTY(10),PERCNT(22,20)
        REAL    LD,LC
        CHARACTER*1     SFLAG
C
   799  WRITE(*,800)
   800  FORMAT(25(/))
        WRITE(*,801)
   801  FORMAT(1X,'DISTRIBUTION           INFORMATION:',//,
      &5X,'1.     DISPLAY    DURATION/COST      DIST.'S',/,
      &5X,'2.     DISPLAY    CORRELATION/REGRESSION          DATA',/,
      &5X,'3.     DISPLAY    PERCENTILES',/,
      &5X,'4.     DISPLAY    COST    FRACTIONS     AT   KEY   EVENTS',/,
      &5X,'5.     --RETURN     TO   MAIN    MENU--',/,
```

```
     &5X,'6.    DISTRIBUTION    INFORMATION',//,
     &5X,'7.    EXIT   PROGRAM',//,
     &1X,'ENTER    CHOICE',///)
C
      READ(*,*)    NV
      GOTO   (810,850,870,900,804,806,807),NV
C
C---RETURN    TO  MAIN   MENU
C
  804  RETURN
C
C---CALL    SUBROUTINE    ON  DISTRIBUTION     INFORMATION
C
  806  CALL   DTHELP
       WRITE(*,800)
       GOTO   799
C
C---EXIT    PROGRAM
C
  807  STOP
C
C---DISPLAY     COST   FRACTIONS    AT  KEY   EVENTS
C
  900  WRITE(*,800)
       WRITE(*,903)
  903  FORMAT(5X,'THE      SIMULATION     RESULTED    IN  THE   FOLLOWING    FRACTIONS',/
     *,5X,'OF    COST   AT  THE   KEY   EVENTS:',//,
     *20X,'KEY    EVENT              FRACTION    OF  COST',//)
       ITEMP=0
       WRITE(*,904)     ITEMP,STPERC
  904  FORMAT(20X,I5,15X,F5.2)
       DO  901   I=1,NKEYEN
          WRITE(*,902)      IKEY(I),FRACN(I)
  902     FORMAT(20X,I5,15X,F5.2)
  901  CONTINUE
       PAUSE
       WRITE(*,800)
       GOTO   799
C
C---WRITE    EMPIRICAL    DENSITIES    TO  SCREEN
C
  810  WRITE(*,800)
       CALL   DWRITE
       PAUSE
       GOTO   799
C
C---WRITE    INFORMATION     ON  CORRELATION     AND   LEAST   SQUARES    ESTIMATORS
C
  850  WRITE(*,800)
       WRITE(*,851)
  851  FORMAT(1X,'INPUT      NUMBER    OF  KEY   EVENT   TO  DISPLAY    CORRELATION',/,
     &1X,'LEAST    SQUARES    DATA   (0  FOR   TOTAL   COST   vs.  PROJECT    DURATION)',/)
       READ(*,*)     NKEY
```

```
      IF (NKEY.EQ.0)   GOTO  860
      IF (NKEY.LT.IKEY(1).OR.NKEY.GT.IKEY(NKEYEN))        THEN
        WRITE(*,*)   'OUT  OF  RANGE  ---  RETURN  TO  MAIN'
        RETURN
      ENDIF
C
      DO 252  I=1,NKEYEN
        IF (IKEY(I).EQ.NKEY)      NNN=I
  252 CONTINUE
C
      IF (CORR(NNN).EQ.-9999.)       THEN
        WRITE(*,800)
        WRITE(*,856)
  856   FORMAT(5X,'NETWORK      IS  DETERMINISTIC!!!',//
     &5X,'THIS    ANALYSIS    IS  NOT  APPLICABLE.')
        PAUSE
        GOTO  799
      ELSE
        WRITE(*,800)
        WRITE(*,852)     IKEY(NNN)
  852   FORMAT(20X,'*****      KEY  EVENT',I5,'     *****',////)
        WRITE(*,853)     BHAT0(NNN),BHAT1(NNN),CORR(NNN)
  853   FORMAT(10X,'SAMPLE       REGRESSION     FUNCTION:',//,
     &15X,'COST    =  ',F12.2,'   +  ',F12.2,'   *  DURATION',///,
     &10X,'SAMPLE     CORRELATION     COEFFICIENT=     ',F4.2,///)
        PAUSE
        GOTO  799
      ENDIF
C
  860 IF (CORR(11).EQ.-9999.)       THEN
        WRITE(*,800)
        WRITE(*,856)
        PAUSE
        GOTO  799
      ELSE
        WRITE(*,800)
        WRITE(*,854)
  854   FORMAT(20X,'*****TOTAL        COST  vs.  PROJECT   DURATION*****',////)
        WRITE(*,855)     BHAT0(11),BHAT1(11),CORR(11)
  855   FORMAT(10X,'SAMPLE       REGRESSION     FUNCTION:',//,
     &15X,'TOTAL     COST   =  ',F10.2,'    +  ',F10.2,'    *  PROJECT    DURATION',///,
     &10X,'SAMPLE     CORRELATION     COEFFICIENT=     ',F4.2,///)
        PAUSE
        GOTO  799
      ENDIF
C
C---DETERMINE     PERCENTILES    OF  DISTRIBUTIONS     AND  WRITE   TO  SCREEN
C
  870 IF (IFLAGG.EQ.0)     CALL   PRCNTL(IFLAGG)
  875 WRITE(*,800)
      WRITE(*,876)
  876 FORMAT(5X,'THIS      PORTION    OF  THE   PROGRAM   WILL   DISPAY',/,
     &2X,'PERCENTILES      OF  THE   DISTRIBUTIONS.',///)
```

```
C
      CALL   PWRITE
      GOTO   799
C
      RETURN
      END
C-------------------------------------------------------------
C
C     SUBROUTINE     D T H E L P
C
C     THIS  SUBROUTINE   PROVIDES   INFORMATION   ON  THE  EMPIRICAL   DISTRIBU-
C     TIONS  OF  THE  KEY  EVENTS  DETERMINED   IN  PROGRAM   AN-COST.
C
C-------------------------------------------------------------
C
      SUBROUTINE    DTHELP
C
  100 WRITE(*,1)
    1 FORMAT(25(/))
      WRITE(*,2)
    2 FORMAT(5X,'CHOOSE      INFORMATION     OPTION',//,
     *10X,'1.     REALIZATION     TIME  & COST  DISTRIBUTIONS',/,
     *10X,'2.     SAMPLE   REGRESSION    FUNCTION    & CORRELATION     COEF.',/,
     *10X,'3.     DISTRIBUTION     PERCENTILES',/,
     *10X,'4.     COST   FRACTIONS    AT  KEY   EVENTS',/,
     *10X,'5.     RETURN   TO  DISTRIBUTION     MENU',//)
      READ(*,*)    NC
C
      IF  (NC.EQ.5)    RETURN
      IF  (NC.LT.1.OR.NC.GT.4)        THEN
        WRITE(*,*)     'OUT   OF   RANGE   ---   RETURN    TO  DIST.   MENU'
        RETURN
      ENDIF
C
      GOTO   (10,20,30,40),NC
C
   10 WRITE(*,1)
      WRITE(*,11)
   11 FORMAT(5X,'THE      REALIZATION     TIME   AND   COST   DISTRIBUTIONS      WERE',/,
     *5X,'DETERMINED      VIA  MONTE   CARLO   SAMPLING    BY  PROGRAM    AN-COST.',//,
     *5X,'THE     COST   DISTRIBUTIONS     ARE  THE   COST   OF  THE   SUBGRAPHS    OF',/,
     *5X,'THE     KEY  EVENTS.     IF  THERE   WERE   COMMON    ACTIVITIES     TO  MORE',/,
     *5X,'THAN     ONE  KEY   EVENT,    IT'S   COST   WAS   SEPARATED    BY  AN  ALGO-',/,
     *5X,'RITHM     THAT   ASSIGNED    MORE   WEIGHT    TO  THE   KEY   EVENT    THAT',/,
     *5X,'WAS    REALIZED     EARLIEST    AND   HAD   LOWER    VARIANCE.',///)
      PAUSE
      GOTO   100
C
   20 WRITE(*,1)
      WRITE(*,21)
   21 FORMAT(5X,'THE      SAMPLE    REGRESSION     FUNCTION    IS  DETERMINED     VIA',/,
     *5X,'LEAST     SQUARES,    AND   THE   SAMPLE   CORRELATION     COEFFICIENT      IS',/,
     *5X,'CALCULATED      IN  THE   STANDARD    MANNER    OVER   THE   INDIVIDUAL',/,
```

```
     *5X,'NETWORK REALIZATIONS IN THE MONTE CARLO SIMULATION.',///,
     *5X,'THESE VALUES WILL GIVE AN INDICATION OF THE RELATIONSHIP',/,
     *5X,'BETWEEN THE COST AND REALIZATION TIME OF THE KEY EVENTS,',/,
     *5X,'AND MAY BE HELPFUL WHEN PREPARING A BID PACKAGE.',///)
      PAUSE
      GOTO 100
C
   30 WRITE(*,1)
      WRITE(*,31)
   31 FORMAT(5X,'THE PERCENTILES OF THE DISTRIBUTIONS ARE CALCULATED',/,
     *5X,'BY LINEAR INTERPOLATION OF THE CUMULATIVE DISTRIBUTION',/,
     *5X,'FUNCTIONS.',//,
     *5X,'A VALUE OF -1 INDICATES THAT A VALUE COULD NOT BE PROPERLY',/,
     *5X,'COMPUTED BECAUSE THE RANGE OF THE EMPIRICAL DISTRIBUTION',/,
     *5X,'WAS TOO NARROW',///)
      PAUSE
      GOTO 100
C
   40 WRITE(*,1)
      WRITE(*,41)
   41 FORMAT(5X,'THE FRACTION OF TOTAL PROJECT COST THAT IS RECEIVED',/,
     *5X,'AT EACH KEY EVENT IS THE AVERAGE OVER ALL MONTE CARLO SAM-',/,
     *5X,'PLES. NOTE: THIS FRACTION IS AFTER THE PERCENTAGE IS ',/,
     *5X,'TAKEN OUT TO RECEIVE AT THE PROJECT START.',///)
      PAUSE
      GOTO 100
C
      RETURN
      END
C-----------------------------------------------------------------
C
C     SUBROUTINE D W R I T E
C
C     THIS SUBROUTINE WRITE'S THE DISTRIBUTIONS TO THE SCREEN
C
C-----------------------------------------------------------------
C
      SUBROUTINE DWRITE
C
      COMMON/CFMV/SDISBUR(100),PAY(10,100),RMEAN(11),PINIT,MXPERD,NKEYEN
     *,IKEY(10),TPAY(100),DMEAN(10),DF(22,102),LL(22),NRR(22),NCELLS(22)
     *,WIDTH(22),DSTD(10),RSTD(11),CDF(22,102),CORR(11),BHAT0(11)
     *,BHAT1(11),FRACN(10)
      COMMON/DC'/COST(11),DURN(10),STPERC,PENALTY(10),PERCNT(22,20)
      REAL LC,LD,RC,RD
      CHARACTER*1 SFLAG
C
      WRITE(*,811)
  811 FORMAT(1X,'INPUT NUMBER OF KEY EVENT TO DISPLAY DIST. FUNC.'S',
     &1X,'(0, FOR TOTAL COST)',/)
      READ(*,*) NKEY
      IF (NKEY.EQ.0) GOTO 830
      IF (NKEY.LT.IKEY(1).OR.NKEY.GT.IKEY(NKEYEN)) THEN
```

```
          WRITE(*,*)     'OUT  OF  RANGE   ---  RETURN   TO  MENU'
          RETURN
        ENDIF
C
        DO  230   I=1,NKEYEN
          IF  (IKEY(I).EQ.NKEY)       NNN=I
  230   CONTINUE
C
        WRITE(*,800)
  800   FORMAT(25(/))
        WRITE(*,812)
        WRITE(*,813)
  812   FORMAT(20X,'DURATION',28X,'COST')
  813   FORMAT(11X,'LEFT',4X,'RIGHT',4X,'PROB.',14X,'LEFT',4X,'RIGHT',4X,
     &'PROB.',//)
        LD=0
        RD=0
        LC=0
        RC=0
        ILIMIT=MAX(NCELLS(NNN)+2,NCELLS(NNN+10)+2)
        DO  820   I=1,ILIMIT
          IF  (MOD(I,15).EQ.0)       THEN
            WRITE(*,*)     'Hit  Return   for  more.'
            READ(*,893)     SFLAG
  893       FORMAT(A1)
          ENDIF
          IF  (I.EQ.1)    THEN
            WRITE(*,814)       LL(NNN),DF(NNN,I),LL(NNN+10),DF(NNN+10,I)
            LD=LL(NNN)
            RD=LD+WIDTH(NNN)
            LC=LL(NNN+10)
            RC=LC+WIDTH(NNN+10)
            GOTO   820
          ELSE   IF   (I.EQ.ILIMIT)      THEN
            IF   (NCELLS(NNN).EQ.NCELLS(NNN+10))            THEN
              WRITE(*,815)      NRR(NNN),DF(NNN,I),NRR(NNN+10),DF(NNN+10,I)
            ELSE   IF   (NCELLS(NNN).GT.NCELLS(NNN+10))           THEN
              WRITE(*,821)      NRR(NNN),DF(NNN,I)
            ELSE
              WRITE(*,822)      NRR(NNN+10),DF(NNN+10,I)
            ENDIF
            WRITE(*,*)     'Hit   return   to  continue'
            READ(*,893)     SFLAG
            GOTO   820
          ELSE
            IF   (NCELLS(NNN)+2.GT.I.AND.NCELLS(NNN+10)+2.GT.I)              THEN
              WRITE(*,816)      LD,RD,DF(NNN,I),LC,RC,DF(NNN+10,I)
            ELSE   IF   (NCELLS(NNN)+2.EQ.I)         THEN
              WRITE(*,823)      NRR(NNN),DF(NNN,I),LC,RC,DF(NNN+10,I)
            ELSE   IF   (NCELLS(NNN)+2.LT.I)         THEN
              WRITE(*,824)      LC,RC,DF(NNN+10,I)
            ELSE   IF   (NCELLS(NNN+10)+2.EQ.I)         THEN
              WRITE(*,825)      LD,RD,DF(NNN,I),NRR(NNN+10),DF(NNN+10,I)
```

```
            ELSE  IF  (NCELLS(NNN+10)+2.LT.I)        THEN
              WRITE(*,826)      LD,RD,DF(NNN,I)
            ENDIF
            LD=RD
            RD=RD+WIDTH(NNN)
            LC=RC
            RC=RC+WIDTH(NNN+10)
          ENDIF
  820   CONTINUE
  814   FORMAT(11X,'----',2X,I5,5X,F6.4,16X,'----',2X,I5,5X,F6.4)
  815   FORMAT(7X,I5,9X,'----',2X,F6.4,12X,I5,6X,'----                 ',F6.4,//)
  821   FORMAT(7X,I5,9X,'----',2X,F6.4,//)
  822   FORMAT(45X,I5,6X,'----',5X,F6.4,//)
  816   FORMAT(7X,F8.2,2X,F8.2,2X,F6.4,12X,F8.2,2X,F8.2,2X,F6.4)
  823   FORMAT(7X,I5,8X,'----',2X,F6.4,12X,F8.2,2X,F8.2,2X,F6.4)
  824   FORMAT(45X,F8.2,2X,F8.2,2X,F6.4)
  825   FORMAT(7X,F8.2,2X,F8.2,2X,F6.4,12X,I5,5X,'----',2X,F6.4)
  826   FORMAT(7X,F8.2,2X,F8.2,2X,F6.4)
C
C---ECHO   MEANS   AND   STANDARD   DEVIATIONS   OF  DURATION   AND   COST
C
        WRITE(*,818)      DMEAN(NNN),RMEAN(NNN)
        WRITE(*,819)      DSTD(NNN),RSTD(NNN)
  818   FORMAT(/////,5X,'           MEAN   =  ',F10.2,20X,'MEAN           =',F10.2)
  819   FORMAT(5X,'STD      DEV.   =  ',F10.2,20X,'STD      DEV.   =',F10.2,//)
        RETURN
C
C---DISTRIBUTIONS       FOR   TOTAL   COST   DISTRIBUTION
C
  830   WRITE(*,800)
        WRITE(*,832)
        WRITE(*,833)
  832   FORMAT(20X,'COST')
  833   FORMAT(11X,'LEFT',4X,'RIGHT',4X,'PROB.'//)
        LC=0
        RC=0
        ILIMIT=NCELLS(22)+2
        DO  840   I=1,ILIMIT
          IF  (MOD(I,15).EQ.0)       THEN
            WRITE(*,*)      'Hit   Return    for   more.'
            READ(*,893)      SFLAG
          ENDIF
          IF  (I.EQ.1)     THEN
            WRITE(*,834)      LL(22),DF(22,I)
            LC=LL(22)
            RC=LC+WIDTH(22)
            GOTO   840
          ELSE   IF   (I.EQ.ILIMIT)      THEN
            WRITE(*,835)      NRR(22),DF(22,I)
            WRITE(*,*)      'Hit    return    to   continue'
            READ(*,893)      SFLAG
            GOTO   840
          ELSE
```

```
            WRITE(*,836)      LC,RC,DF(22,I)
            LC=RC
            RC=RC+WIDTH(22)
          ENDIF
  840   CONTINUE
  834   FORMAT(8X,'----',5X,I5,5X,F6.4)
  835   FORMAT(7X,I5,9X,'----',2X,F6.4,//)
  836   FORMAT(7X,F8.2,2X,F8.2,2X,F6.4)
C
C---ECHO   MEAN  AND  STANDARD   DEVIATION   OF  TOTAL   COST
C
        WRITE(*,838)     RMEAN(11),RSTD(11)
  838   FORMAT(/////,5X,'          MEAN   =  ',F10.2,7X,'STD.      DEV.   =',F10.2)
C
        RETURN
        END

C-------------------------------------------------------------------
C
C         SUBROUTINE      H  E  L  P  M  A  I  N
C
C         THIS  SUBROUTINE    PROVIDES    GENERAL    INFORMATION    ABOUT   PROGRAM   CASH
C
C-------------------------------------------------------------------
C
        SUBROUTINE    HELPMAIN
C
        WRITE(*,1)
      1 FORMAT(25(/))
        WRITE(*,2)
      2 FORMAT(15X,'P      R  O  G  R  A  M       C  A  S  H',///,
       *5X,'THIS    PROGRAM    IS  DESIGNED    TO  ASSIST    THE   PROJECT    MANAGER    IN',/,
       *5X,'DETERMINING       A  BID   PACKAGE    FOR  PROJECTS    THAT   CAN   BE  MODELED',/,
       *5X,'BY    DIRECTED     ACYCLIC     NETWORKS.',///)
        PAUSE
        WRITE(*,3)
      3 FORMAT(///5X,'BEFORE        THIS   PROGRAM    CAN  BE  RUN,   YOU   NEED   TO  RUN',/,
       *5X,'PROGRAM     AN-COST.     AN-COST    WILL    CREATE    FILE   CASHFLOW.DAT',/,
       *5X,'WITH     THE   NECESSARY     DATA.',///)
        PAUSE
        WRITE(*,1)
        WRITE(*,4)
      4 FORMAT(5X,'THIS      PROGRAM    ALLOWS    THE   FOLLOWING:',//,
       *10X,'1.     VIEWING    OF  CASH   FLOW   ARRAYS',/,
       *10X,'2.     VIEWING    OF  KEY   EVENT   DISTRIBUTION     INFORMATION',/,
       *10X,'       ---REALIZATION       TIME   DISTRIBUTION',/,
       *10X,'       ---COST    DISTRIBUTION',/,
       *10X,'       ---REGRESSION      FUNCTION    AND   CORRELATION      COEF.',/,
       *10X,'           (DURATION     vs.   COST)',/,
       *10X,'       ---PERCENTILES      OF   DISTRIBUTIONS',/,
       *10X,'       ---FRACTION     OF  PROJECT    COST   AT   KEY   EVENTS',/,
       *10X,'3.     DETERMINISTIC      CASH   FLOW   CALCULATIONS',/,
       *10X,'4.     PROBABILISTIC      CASH   FLOW   CALCULATIONS',///,
```

```
     *5X,'NOTE:       FOR   MORE   INFORMATION     ON   ANY   OF   THESE    TOPICS,',/,
     *5X,'            CHOOSE    INFORMATION      ON   SUB-MENUS.',//)
      PAUSE
      WRITE(*,1)
C
      RETURN
      END
C--------------------------------------------------------------------
C
C     SUBROUTINE      I  N  P  U  T
C
C     THIS   SUBROUTINE    OBTAINS    NEEDED    INPUT    VALUES
C
C--------------------------------------------------------------------
C
      SUBROUTINE      INPUT(NOPTION,ISTOP,IFLAGG)
C
      COMMON/CFMV/SDISBUR(100),PAY(10,100),RMEAN(11),PINIT,MXPERD,NKEYEN
     *,IKEY(10),TPAY(100),DMEAN(10),DF(22,102),LL(22),NRR(22),NCELLS(22)
     *,WIDTH(22),DSTD(10),RSTD(11),CDF(22,102),CORR(11),BHAT0(11)
     *,BHAT1(11),FRACN(10)
      COMMON/CFP/ALPHA,BETA,POWER,RATE,DESIRE,NCYCLE,CAPITAL,TOL,CMODEL
      COMMON/DCV/COST(11),DURN(10),STPERC,PENALTY(10),PERCNT(22,20)
      INTEGER    CMODEL
      CHARACTER*1      NCHAR
C
      WRITE(*,325)
  325 FORMAT(25(/))
      WRITE(*,300)
  300 FORMAT(5(/),1X,'INPUT         INTEREST     RATE   ON   MONEY    DEPOSITED')
      READ(*,*)     ALPHA
      WRITE(*,301)
  301 FORMAT(5(/),1X,'INPUT         INTEREST     RATE   ON   MONEY    BORROWED')
      READ(*,*)     BETA
      WRITE(*,302)
  302 FORMAT(5(/),1X,'INPUT         RETENTION     RATE')
      READ(*,*)     RATE
      IF(NOPTION.EQ.3)        THEN
        WRITE(*,303)
  303   FORMAT(5(/),1X,'INPUT         DESIRED     PROFIT    PERCENTAGE')
        READ(*,*)     DESIRE
      ENDIF
      WRITE(*,304)
  304 FORMAT(5(/),1X,'INPUT         INITIAL     CAPITAL')
      READ(*,*)     CAPITAL
      IF  (NOPTION.EQ.3)        THEN
        WRITE(*,305)
  305   FORMAT(5(/),1X,'INPUT         TOLERANCE     LEVEL    FOR    BISECTION     METHOD')
        READ(*,*)     TOL
  310   WRITE(*,306)
  306   FORMAT(5(/),1X,'INPUT         PROFIT    MODEL    THAT    YOU    WANT    TO   BE   USED',//,
     &5X,'1.      PROFIT    =  f(CAPITAL)',/,
     &5X,'2.      PROFIT    =  f(PROJECT      COST)',//)
```

```
          READ(*,*)    CMODEL
          IF  (CMODEL.LT.1.OR.CMODEL.GT.2)          GOTO  310
        ENDIF
C
C---IF   DETERMINISTIC     PROBLEM,    RETURN
C
        IF  (NJPTION.EQ.3)      RETURN
C
        WRITE(*,319)
  319   FORMAT(//,1X,'YOU      WILL   NOW   BE  ASKED   TO  INPUT   THE  COST   THAT',/,
     &1X,'YOU    WILL   --QUIRE    AT  THE   KEY   NODES,   AND   THE  DELIVERY',/,
     &1X,'DATES     OF  THE   KEY   NODES.',//,
     &5X,'NOTE:      INPUT    THE   COST   FOR   THE   ENTIRE   SUBGRAPH    OF  THE',/,
     &5X,'           KEY   EVENT.    THE  PROGRAM   WILL   REMOVE   THE  ALLOTTED',/,
     &5X,'           PERCENTAGE     TO  RECEIVE    AT  THE   PROJECT    START.',//)
        PAUSE
C
        DO  320   II=1,NKEYEN
  326     WRITE(*,325)
          WRITE(*,330)     IKEY(II)
  330     FORMAT(/20X,'FOR       KEY   EVENT    ',I3,'.....',////)
          WRITE(*,327)
  327   FORMAT(5X,'DO      YOU   WANT   TO  EXAMINE    THE   FOLLOWING    BEFORE',/,
     &5X,'PROCEEDING      WITH   THE   INPUT:',///,
     &10X,'1.      EMPIRICAL     DENSITIES',/,
     &10X,'2.      PERCENTILES      OF  DISTRIBUTIONS',/,
     &10X,'3.      PROCEED     WITH   INPUT',/,
     &10X,'4.      ---RETURN     TO  MAIN   MENU---',//)
          READ(*,*)     IVAL
          IF  (IVAL.EQ.4)     RETURN
          IF  (IVAL.LT.1.OR.IVAL.GT.3)          THEN
            WRITE(*,*)      'VALUE    OUT   OF  RANGE    ---  RETURN    TO  MAIN   MENU'
            RETURN
          ENDIF
          IF  (IVAL.EQ.1     )  THEN
            CALL   DWRITE
            GOTO   326
          ELSE   IF  (IVAL.EQ.2)     THEN
            IF  (IFLAGG.EQ.0)      CALL   PRCNTL(IFLAGG)
            CALL   PWRITE
            GOTO   326
          ENDIF
C
          WRITE(*,321)      IKEY(II)
  321   FORMAT(//,1X,'INPUT         DURATION    AND   COST   DECIDED    ON  AT  KEY   NODE',I3)
          READ(*,*)     DURN(II),COST(II)
          WRITE(*,322)      IKEY(II)
  322   FORMAT(//,1X,'INPUT         THE  LATE    PENALTY     FOR   KEY   EVENT',I4,//,
     &,1X,'PER      PERIOD',///)
          READ(*,*)     PENALTY(II)
  320   CONTINUE
C
        TESUM=0.0
```

```
      DO  323   I=1,NKEYEN
        TESUM=TESUM+COST(I)
  323 CONTINUE
      COST(11)=TESUM*STPERC
C
      RETURN
      END
C-----------------------------------------------------------------
C
C         SUBROUTINE       I  N  T  R  T
C
C             THIS   SUBROUTINE     CALCULATES     INTEREST    IN   EACH   PERIOD
C
C-----------------------------------------------------------------
C
      SUBROUTINE     INTRT(I,CASH,POWER,ALPHA,BETA,TIAF,TIBA,TRETEN,FLAG2)
C
      INTEGER     I,FLAG2
      REAL        CASH,POWER,ALPHA,BETA,TIAF,TIBA,TRETEN
C
      REAL        GA,GB
C
      TRETEN=TRETEN*(1+ALPHA)**POWER
      IF  (CASH.GE.0.00)       THEN
        GA=CASH*(1+ALPHA)**POWER-CASH
        TIAF=TIAF+GA
        CASH=CASH+GA
        IF  (FLAG2.EQ.1)       THEN
          WRITE(3,1300)       I,GA,CASH
 1300     FORMAT(3X,'I       =',I5,5X,'IAF       =',F10.3,5X,'CASH        =',F10.3)
        ENDIF
      ELSE
        GB=-1.0*CASH*(1+BETA)**POWER+CASH
        TIBA=TIBA+GB
        CASH=CASH-GB
        IF  (FLAG2.EQ.1)       THEN
          WRITE(3,1305)       I,GB,CASH
 1305     FORMAT(8X,'I       =',I5,5X,'IBA       =',F10.3,5X,'CASH        =',F10.3)
        ENDIF
      ENDIF
      RETURN
      END
C-----------------------------------------------------------------
C
C     SUBROUTINE       P  C  F  H  E  L  P
C
C     THIS   SUBROUTINE     PROVIDES    INFORMATION     ON   THE   CASH   FLOW   CALCULATIONS
C     IN  A  PROBABILISTIC      PROBLEM.
C
C-----------------------------------------------------------------
C
      SUBROUTINE     PCFHELP
C
```

```
      WRITE(*,1)
    1 FORMAT(25(/))
      WRITE(*,2)
    2 FORMAT(5X,'THIS      PORTION   OF  THE  PROGRAM  DOES  CASH  FLOW  CALCULA-',/,
     *5X,'TIONS   IN  THE  CASE  OF  A  PROBABILISTIC   NETWORK.',//,
     *5X,'YOU   WILL  BE  ASKED  TO  INPUT:',//,
     *10X,'1.    INTEREST  RATE  ON  MONEY  DEPOSITED',/,
     *10X,'2.    INTEREST  RATE  ON  MONEY  BORROWED',/,
     *10X,'3.    RETENTION   RATE',/,
     *10X,'4.    INITIAL   CAPITAL',//,
     *10X,'5.    AMOUNT  OF  MONEY  TO  BE  RECEIVED   AT  EACH  KEY  EVENT',/,
     *10X,'6.    DUE  DATE  OF  EACH  KEY  EVENT',/,
     *10X,'7.    LATE  PENALTY  (PER  PERIOD)  FOR  EACH  KEY  EVENT',//)
      PAUSE
C
      WRITE(*,10)
   10 FORMAT(5(/))
      WRITE(*,3)
    3 FORMAT(5X,'THE      DISTRIBUTION    INFORMATION    WILL  BE  AVAILABLE   TO',/,
     *5X,'YOU   WHEN  INPUTTING   5-7,  ABOVE.',///,
     *5X,'THE   RESULT  WILL  BE  AN  EXPECTED   PROFIT  FOR  THE  GIVEN  VALUES',/,
     *5X,'OF   1-7,  ABOVE.',//,
     *5X,'YOU   MAY  RUN  AS  MANY  ALTERNATIVES   AS  YOU  LIKE  TO  DETERMINE',/,
     *5X,'THE   BID  PACKAGE  THAT  YOU  WILL  MAKE.  THE  OUTPUT  WILL  BE  ',/,
     *5X,'SENT   TO  THE  SCREEN,  AND  ALSO  TO  A  FILE  NAMED  "CASHOUT.OUT"',/,
     *5X,'WHICH   WILL  BE  ON  THE  DEFAULT  DRIVE.',///)
      PAUSE
C
      RETURN
      END
C-----------------------------------------------------------------
C
C     SUBROUTINE     P R C N T L
C
C     THIS  SUBROUTINE   CALCULATES  THE  PERCENTILES   OF  THE  DISTRIBUTIONS.
C     IT  USES  A LINEAR  INTERPOLATION   BETWEEN  ENDPOINTS   OF  THE  EMPIRICAL
C     DISTRIBUTIONS.
C
C-----------------------------------------------------------------
C
      SUBROUTINE    PRCNTL(IFLAGG)
      COMMON/DCV/COST(11),DURN(10),STPERC,PENALTY(10),PERCNT(22,20)
      COMMON/CFMV/SDISBUR(100),PAY(10,100),RMEAN(11),PINIT,MXPERD,NKEYEN
     *,IKEY(10),TPAY(100),DMEAN(10),DF(22,102),LL(22),NRR(22),NCELLS(22)
     *,WIDTH(22),DSTD(10),RSTD(11),CDF(22,102),CORR(11),BHAT0(11)
     *,BHAT1(11),FRACN(10)
C
      WRITE(*,*)     'WORKING'
      IFLAGG=1
      DO  2000   I=1,22
      DO  2005   J=1,20
        PERCNT(I,J)=0.0
 2005 CONTINUE
```

```
 2000   CONTINUE
C
        MARK1=1
        MARK2=0
C
C---DETERMINE      PERCENTILES     THAT   ARE   MULTIPLES    OF   .05
C
 2006   DO  2008   IN=MARK1,NKEYEN+MARK2
        DO  2010   IP=1,20
          RIP=IP*5/100.0
        DO  2012   ID=1,NCELLS(IN)+2
          IF  (ID.EQ.1.AND.CDF(IN,ID).GT.RIP)            THEN
            PERCNT(IN,IP)=-1
            GOTO   2010
          ENDIF
          IF  (CDF(IN,NCELLS(IN)+2).LT.RIP.AND.ID.EQ.NCELLS(IN)+2)                  THEN
            PERCNT(IN,IP)=-1
            GOTO   2010
          ENDIF
          IF  (CDF(IN,ID).LT.RIP)        GOTO   2012
          IF  (CDF(IN,ID).EQ.RIP)        THEN
            PERCNT(IN,IP)=LL(IN)+WIDTH(IN)*(ID-1)
            GOTO   2010
          ENDIF
          SLOPE=(CDF(IN,ID)-CDF(IN,ID-1))/WIDTH(IN)
          B=CDF(IN,ID)-SLOPE*(LL(IN)+WIDTH(IN)*(ID-1))
          IF  (SLOPE.EQ.0.)      THEN
            PERCNT(IN,IP)=LL(IN)+WIDTH(IN)*(ID-1)
          ELSE
            PERCNT(IN,IP)=(RIP-B)/SLOPE
          ENDIF
          GOTO   2010
 2012   CONTINUE
 2010   CONTINUE
 2008   CONTINUE
C
        IF  (MARK1.EQ.1)      THEN
          MARK1=11
          MARK2=10
          GOTO   2006
        ENDIF
C
        IF  (MARK1.EQ.11)      THEN
          MARK1=22
          MARK2=22-NKEYEN
          GOTO   2006
        ENDIF
C
        RETURN
        END
C----------------------------------------------------------------------
C
C       SUBROUTINE      P R O B A B
```

```
C
C       THIS  SUBROUTINE   DOES  CASH  FLOW  CALCULATIONS    IF  THE  NETWORK   IS
C       PROBABILISTIC.     NOTE:   IN  THIS  CASE  IT  SIMPLY  CALCULATES    THE
C       EXPECTED   PROFIT  FOR  THE  VALUES  WHICH  ARE  GIVEN  IN  THE  BID  PACKAGE
C
C-------------------------------------------------------------------------
C
      SUBROUTINE   PROBAB(NOPTION,IFLAGG)
C
      COMMON/CFMV/SDISBUR(100),PAY(10,100),RMEAN(11),PINIT,MXPERD,NKEYEN
     *,IKEY(10),TPAY(100),DMEAN(10),DF(22,102),LL(22),NRR(22),NCELLS(22)
     *,WIDTH(22),DSTD(10),RSTD(11),CDF(22,102),CORR(11),BHATO(11)
     *,BHAT(11),FRACN(10)
      COMMON/CFP/ALPHA,BETA,POWER,RATE,DESIRE,NCYCLE,CAPITAL,TOL,CMODEL
      COMMON/DCV/COST(11),DURN(10),STPERC,PENALTY(10),PERCNT(22,20)
      INTEGER    NPERD(10),FLAG2,CMODEL
      REAL       TIAF,TIBA,TRETEN,RETEN,RATIO(10),DUMPAY(100)
C
 1549 WRITE(*,1550)
 1550 FORMAT(25(/))
      WRITE(*,1551)
 1551 FORMAT(////10X,'1.        PROCEED   WITH   CASH   FLOW   CALCULATIONS',/,
     *10X,'2.     PROBABILISTIC     CASH   FLOW   INFORMATION',/,
     *10X,'3.     RETURN   TO  MAIN   MENU',///)
      WRITE(*,1552)
 1552 FORMAT(5X,'ENTER       CHOICE')
      READ(*,*)     ICHOICE
      IF  (ICHOICE.EQ.2)      THEN
        CALL   PCFHELP
        GOTO   1549
      ENDIF
      IF  (ICHOICE.EQ.3)      RETURN
C
      ISTOP=0
      CALL   INPUT(NOPTION,ISTOP,IFLAGG)
      IF  (ISTOP.EQ.1)      RETURN
C
      POWER=NCYCLE/365.0
      DESI=DESIRE/100.0
      TPINIT=0.0
C
C---COMPUTE     THE  RATIO   OF  WHAT  WE  ASK  FOR  AT  A  KEY  EVENT   AND  WHAT   THE
C---EXPECTED     COST   IS  OF  THE  SUBGRAPH   OF  THE  KEY  EVENT.    ALSO  CALCULATE
C---THE   RATIO   OF  THE  COST  WE  WANT  TO  RECEIVE   AT  THE  PROJECT   START   TO
C---WHAT    THE  EXPECTED    VALUE   IS
C
      DO  1590   JJ=1,100
        DUMPAY(JJ)=0.0
 1590 CONTINUE
      DO  1591   JJ=1,NKEYEN
        RATIO(JJ)=0.0
        NPERD(JJ)=0.0
 1591 CONTINUE
```

```
C
      DO  1600   JJ=1,NKEYEN
        RATIO(JJ)=COST(JJ)/RMEAN(JJ)
 1600 CONTINUE
      IF  (PINIT.GT.0.)      THEN
        RATIO(11)=COST(11)/PINIT
        TPINIT=RATIO(11)*PINIT
      ENDIF
C
C---ALTER    INCOME    ARRAYS    ACCORDINGLY
C
      DO  1605   J=1,NKEYEN
      DO  1610   JJ=1,MXPERD
        PAY(J,JJ)=PAY(J,JJ)*RATIO(J)
 1610 CONTINUE
 1605 CONTINUE
C
C---DETERMINE     TOTAL    INCOME    IN   ALL   PERIODS    FOR   ALL   KEY   EVENTS
C
      DO  1616   I=1,MXPERD
      DO  1617   J=1,NKEYEN
        DUMPAY(I)=DUMPAY(I)+PAY(J,I)
 1617 CONTINUE
 1616 CONTINUE
C
C---CALCULATE     THE   PERIOD    IN  WHICH   WE  PROMISE    DELIVERY    AT  EACH   KEY   EVENT
C
      DO  1615   JJ=1,NKEYEN
        IF((DURN(JJ)/NCYCLE).EQ.INT(DURN(JJ)/NCYCLE))                  THEN
            NPERD(JJ)=DURN(JJ)/NCYCLE
        ELSE
            NPERD(JJ)=DURN(JJ)/NCYCLE+1
        ENDIF
 1615 CONTINUE
C
        CASH=0.0
        TIAF=0.00
        TIBA=0.00
        TRETEN=0.00
        FLAG2=1
      DO  1510   I=1,MXPERD+1
C
C---INITIAL      PERIOD
C
       IF   (I.EQ.1)     THEN
         CASH=CAPITAL+TPINIT-SDISBUR(I)
         CALL    INTRT(I,CASH,POWER,ALPHA,BETA,TIAF,TIBA,TRETEN,FLAG2)
         WRITE(*,1950)        I,CASH
 1950    FORMAT(1X,'PERIOD',I3,2X,'CASH              =',F10.2)
         GO  TO  1510
       ENDIF
C
C---AFTER    LAST   PERIOD
```

```
C
        IF  (I.EQ.MXPERD+1)       THEN
         CASH=CASH+DUMPAY(I-1)+TRETEN
         DO  1520   JJ=1,NKEYEN
           IF  (NPERD(JJ).LT.I)       CASH=CASH-PENALTY(JJ)*CUM(JJ,I)
 1520    CONTINUE
         CTERM=CASH
         WRITE(*,1991)      CASH
         WRITE(3,1991)      CASH
 1991    FORMAT(25X,'TERMINAL         CASH   POSITION    =',F10.3)
         GO  TO  1510
        END  IF
C
C---ALL   PERIODS   BETWEEN   BEGINNING    AND   END
C
         RETEN=DUMPAY(I-1)*RATE
         TRETEN=TRETEN+RETEN
         CASH=CASH+DUMPAY(I-1)*(1-RATE)-SDISBUR(I)
         DO  1530   JJ=1,NKEYEN
           IF  (NPERD(JJ).LT.I)       CASH=CASH-PENALTY(JJ)*CUM(JJ,I)
 1530    CONTINUE
         CALL   INTRT(I,CASH,POWER,ALPHA,BETA,TIAF,TIBA,TRETEN,FLAG2)
         WRITE(*,1980)      TRETEN,RETEN
 1980    FORMAT(/15X,'CUMULATIVE          RETENTION     =',F10 3,3X,'RETENTION=',
     *          F9.3/)
         WRITE(*,1951)      I,CASH
 1951    FORMAT(1X,'PERIOD',I3,2X,'CASH            =',F10.2)
C
C    ----------------------------
C
 1510  CONTINUE
       PROFIT=(CASH-CAPITAL)/RMEAN(11)
C
       WRITE(*,1901)
       WRITE(3,1901)
 1901  FORMAT(//5X,'---------BID          PACKAGE---------',//)
       WRITE(*,1902)       COST(11)
       WRITE(3,1902)       COST(11)
 1902  FORMAT(1X,F10.2,'RECEIVED            AT   PROJECT    START')
       DO   1910    I=1,NKEYEN
         WRITE(*,1911)       IKEY(I),COST(I),DURN(I),PENALTY(I)
         WRITE(3,1911)       IKEY(I),COST(I),DURN(I),PENALTY(I)
 1910  CONTINUE
 1911  FORMAT(/10X,'KEY        EVENT',I5,/,
     *          BID       LEVEL            =',F10.2,/,
     *          DELIVERY         DATE   =',F10.2,/,
     *          A             PENALTY      =',F10.2,///)

       W                    PROFIT
       W                    PROFIT
                                         PROFIT    =',F12.3/)
       W                    RMEAN(11)
                            RMEAN(11)
```

```
1903  FORMAT(//10X,'EXPECTED         PROJECT    COST    =  ',F12.2,//)
      WRITE(*,1904)
      WRITE(3,1904)
1904  FORMAT(/2X,'UNDER       THE   CONSTRAINTS:      ')
      WRITE(*,1905)       CAPITAL
      WRITE(3,1905)       CAPITAL
1905  FORMAT(/8X,'INITIAL         CAPITAL     =',F10.2)
      WRITE(*,1906)       ALPHA
      WRITE(*,1907)       BETA
      WRITE(*,1908)       RATE
      WRITE(3,1906)       ALPHA
      WRITE(3,1907)       BETA
      WRITE(3,1908)       RATE
1906  FORMAT(/8X,'INTEREST         RATE    ON    MONEY    DEPOSIT     =',F6.4,'%')
1907  FORMAT(/8X,'INTEREST         RATE    ON    MONEY    BORROWT     =',F6.4,'%')
1908  FORMAT(/8X,'RETENTION         RATE    =  ',F6.4,'%')
        RETURN
        END
C----------------------------------------------------------------
C
C       SUBROUTINE       P  W  R  I  T  E
C
C       THIS   SUBROUTINE     WRITES    THE   PERCENTILES      TO   THE   SCREEN
C
C----------------------------------------------
C
      SUBROUTINE      PWRITE
      COMMON/DCV/COST(11),DURN(10),STPERC,PENALTY(10),PERCNT(22,2[illegible]
      COMMON/CFMV/SDISBUR(100),PAY(10,100),RMEAN(11),[illegible],MXPERC,NKEYEN
     *,IKEY(10),TPAY(100),DMEAN(10),DF(22,102),[illegible](22),NRR(22),NCF[illegible]S(2[illegible]
     *,WIDTH(22),DSTD(10),RSTD(11),CDF(2[illegible],102),CORR([illegible]),BHAT[illegible]
     *,BHAT1(11),FRACN(10)
C
      WRITE(*,878)
 878  FORMAT(5X,'ENTER        NUMBER    OF    KEY    EVENT    (ENTER       [illegible]
      READ(*,*)      NKEY
      IF   (NKEY.EQ.0)       GOTO    885
      IF   (NKEY.LT.IKEY(1).OR.NKEY.GT.IKEY(NKEYEN))                   THEN
        WRITE(*,*)       'OUT    OF    RANGE          RETURN    [illegible]    MAIN
        RETURN
      ENDIF
C
      DO   231    I=1,NKEYEN
        IF   (IKEY(I).EQ.NKEY)          NNN=[illegible]
 231  CONTINUE
C
 885  IF   (NKEY.EQ.0)       THEN
        WRITE(*,887)
        WRITE(*,888)
      ELSE
        WRITE(*,886)       IKEY(NNN)
        WRITE(*,889)
        WRITE(*,890)
```

```
      ENDIF
 886  FORMAT(20X,'KEY        EVENT    ---',I3)
 889  FORMAT(15X,'DURATION',20X,'COST',/)
 890  FORMAT(5X,'PERCENTILE                VALUE               PERCENTILE           VALUE')
 887  FORMAT(20X,'TOTAL       COST')
 888  FORMAT(15X,'      PERCENTILE                VALUE')
C
      DO  891  I=1,20
        RIP=I*5/100.
        IF  (NKEY.EQ.0)     THEN
          IF  (I.EQ.11)     THEN
            WRITE(*,*)      'Hit   Return    for   more.'
            READ(*,893)      SFLAG
          ENDIF
          WRITE(*,894)      RIP,PERCNT(22,I)
          GOTO   891
        ENDIF
        IF  (I.EQ.11)     THEN
          WRITE(*,*)      'Hit   Return    for   more.'
          READ(*,893)      SFLAG
        ENDIF
        WRITE(*,[illegible])      RIP,PERCNT(NNN,[illegible]),RIP,PERCNT(NNN+[illegible])
 891  CONTINUE
      PAUSE
 893  FORMAT(A[illegible])
 [illegible]  FORMAT([illegible])
 [illegible]  FORMAT([illegible])

      RETURN
      END
```

[illegible]

```
     &10X,'3.      EXPECTED    "OUTFLOW"    vs.   TOTAL   "INFLOW"',/,
     &10X,'4.      --RETURN    TO  MAIN   MENU--',/,
     &10X,'5.      CASH   FLOW   INFORMATION',//,
     &10X,'6.      EXIT   PROGRAM',//)
      READ(*,*)    NV
      IF  (NV.EQ.6)     STOP
      IF  (NV.LT.1.OR.NV.GT.5)          THEN
        WRITE(*,*)     'VALUE    OUT   OF   RANGE'
        WRITE(*,*)     '---RETURN     TO   MAIN---'
        RETURN
      ENDIF
C
      IF  (NV.EQ.4)     RETURN
C
      IF  (NV.EQ.5)     THEN
        CALL   CFHELP
        GOTO   222
      ENDIF
C
      IF  (NV.EQ.2)     THEN
        WRITE(*,*)     'DUE   TO  WHICH   KEY   EVENT?   (0  IF  TOTAL   "INFLOW")'
        READ(*,*)    NKEY
        IF  (NKEY.EQ.0)     GOTO   240
        IF  (NKEY.LT.IKEY(1).OR.NKEY.GT.IKEY(NKEYEN))                 THEN
          WRITE(*,*)     'OUT   OF   RANGE   ---   RETURN    TO   MAIN'
          RETURN
        ENDIF
        DO  230   I=1,NKEYEN
          IF  (IKEY(I).EQ.NKEY)        NNN=I
 230    CONTINUE
      ENDIF
C
 240  IF  (N[illegible]Q.1.OR.NV.EQ.2)          THEN
C
        WRITE(*,290)
 290    FORMAT(///12X,'PERIOD                              FLOW',//)
        JI=0
        IF  (NV.EQ.2.AND.NKEY.EQ.0)          WRITE(*,292)     JI,PINIT
 292    FORMAT(10X,I5,8X,F12.2)
        DO  210   I=1,MXPERD
          IF  (MOD(I,15).EQ.0)        THEN
            WRITE(*,*)     'Hit   Return   for   more.'
            READ(*,295)     NCHAR
          ENDIF
          IF  (NV.EQ.1)     WRITE(*,291)      I,SDISBUR(I)
          IF  (NV.EQ.2.AND.NKEY.NE.0)          WRITE(*,291)      I,PAY(NNN,I)
          IF  (NV.EQ.2.AND.NKEY.EQ.0)          WRITE(*,291)      I,TPAY(I)
 210    CONTINUE
 291    FORMAT(10X,I5,8X,F12.2)
C
      ELSE   IF  (NV.EQ.3)     THEN
C
        WRITE(*,2[illegible])
```

```
250     FORMAT(///12X,'PERIOD                  "OUTFLOW"           "INFLOW"',//)
        JI=0
        WRITE(*,251)      JI,PINIT
251     FORMAT(12X,I4,20X,F11.2)
        DO  255   I=1,MXPERD
          IF  (MOD(I,15).EQ.0)       THEN
            WRITE(*,*)      'Hit   Return    for   more.'
            READ(*,295)      NCHAR
          ENDIF
          WRITE(*,260)      I,SDISBUR(I),TPAY(I)
260       FORMAT(12X,I4,6X,F11.2,3X,F11.2)
255     CONTINUE
      ENDIF
C
      PAUSE
      GOTO   222
295   FORMAT(A1)
299   FORMAT(25(/))
C
      RETURN
      END
C-----------------------------------------------------------------------
```

END

3-87

DTIC